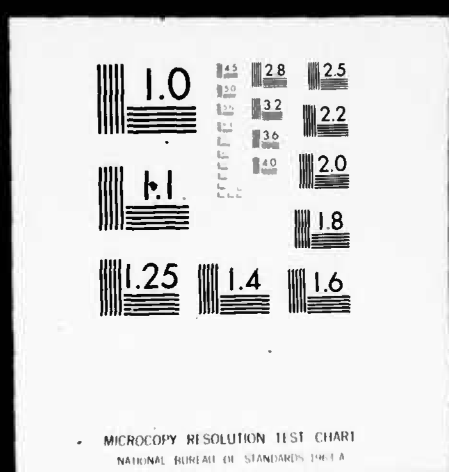
AD-A045 627 GENERAL ELECTRIC CO CINCINNATI OHIO AIRCRAFT ENGINE GROUP F/G 21/5
DEVELOPMENT OF EMISSIONS MEASUREMENT TECHNIQUES FOR AFTERBURNIN--ETC(U)
OCT 75 W C COLLEY, D R FERGUSON, M A SMITH F33615-73-C-2047

UNCLASSIFIED R75AEG459 AFAPL-TR-75-52-SUPPL-2 NL

1 OF 5
ADA045627

1 OF 5

ADA045627



AFAPL-TR-75-52, Supplement 2 R75AEG459

A045491
A019094

AD A045627

DEVELOPMENT OF EMISSIONS MEASUREMENT TECHNIQUES FOR AFTERBURNING TURBINE ENGINES

Supplement 2 - Afterburner Plume Computer Program User's Manual

W.C. Colley
D.R. Ferguson
M.A. Smith

GENERAL ELECTRIC COMPANY
Aircraft Engine Group
Cincinnati, Ohio 45215

DDC
OCT 14 1977

October 1975

Technical Report AFAPL-TR-75-52, Supplement 2

AD No. ___
DDC FILE COPY

Approved for public release;
distribution unlimited

Prepared for

AIR FORCE AERO PROPULSION LABORATORY
Air Force Wright Aeronautical Laboratories
Wright-Patterson Air Force Base, Ohio 45433

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

Leonard C. Angello
Fuels Branch
Fuels and Lubrication Division

FOR THE COMMANDER

Arthur V. Churchill
Chief, Fuels Branch
Fuels and Lubrication Division

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFAPL/SFF *,W-PAFB, OH 45433 to help us maintain a current mailing list".*

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFAPL-TR-75-52, Suppl[redacted]-2	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER: Technical rept.,	
4. TITLE (and Subtitle): DEVELOPMENT OF EMISSIONS MEASUREMENT TECHNIQUES FOR AFTERBURNING TURBINE ENGINES. SUPPLEMENT 2. AFTERBURNER PLUME COMPUTER PROGRAM USER'S MANUAL.	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER: R75AEG459
7. AUTHOR(s): W.C. Colley, D.R. Ferguson [redacted] M.A. Smith	8. CONTRACT OR GRANT NUMBER(s): F33615-73-C-2047
9. PERFORMING ORGANIZATION NAME AND ADDRESS: General Electric Company, Aircraft Engine Group, Cincinnati, Ohio 45215	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: Project 1900
11. CONTROLLING OFFICE NAME AND ADDRESS: Air Force Aero Propulsion Laboratory (AFAPL/SFF), Air Force Wright Aeronautical Laboratories, Wright-Patterson AFB, Ohio 45433	12. REPORT DATE: Oct[redacted]75
	13. NUMBER OF PAGES: ~~457~~ 461 p.
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)

Afterburner Emissions; Environmental Pollution; Exhaust Chemical Kinetics; Exhaust Plume; Exhaust Plume Model; Plume Mixing; Turbine Engine Emissions

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

Computer program "PLUMOD" description and users' manual. PLUMOD is an integral part of the exhaust emissions measurement technique for afterburning turbojet engines developed for AFAPL. It is used in conjunction with gas sample surveys made at the engine exhaust plane to compute the true residual emissions released to the atmosphere. PLUMOD is coded in FORTRAN IV language for the Honeywell 6000 computer.

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

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

403468

PRECEDING PAGE BLANK-NOT FILMED

TABLE OF CONTENTS

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
DISTRIBUTION/AVAILABILITY CODES

SECTION 1.0

INTRODUCTION

Computer program "PLUMOD" described herein is an analytical model of the physical and chemical processes occurring in the mixing and reacting gas flow fields of the exhaust plumes from afterburning turbojet engines. It is an integral part of the exhaust emissions measurement technique developed for such engines for the Air Force Aero Propulsion Laboratory under Contract No. F33615-73-C-2047 (Reference 1).

Determination of the gaseous emissions from afterburning engines by sampling the exhaust gas requires overcoming problems not encountered with nonafterburning engines. These problems are a consequence of the extremely high gas temperatures at the exhaust plane, which can result in continued chemical reactions downstream of this location. If such reactions occur, exhaust plane measurements would yield emissions values which are not representative of the actual contribution to atmospheric pollution. One method of overcoming this problem is to use PLUMOD in conjunction with gas sample surveys made at the engine exhaust plane; PLUMOD predicts analytically the true residual emissions corresponding to the measurements.

This manual contains a summary of the analytical methods, a description of the computer program, and instructions for its use. Full exposition of the analytical methods used in the model, and comparisons of the model's predictions with actual survey measurements made at various axial positions in the exhaust plumes of engines, are contained in Reference 1.

The analytical model is formulated in a building-block manner, incorporating pre-existing elements where applicable. This formulation was expedient in constructing the model, but results in analytical approaches that are sometimes indirect. The most prominent pre-exiting elements are: (a) the steady-state turbulent gas jet mixing analysis used in computer program JETMIX (Reference 2); (b) a formal solution of gas-phase chemical reaction kinetics equations developed by NASA-Lewis Research Center (Reference 3); (c) an approximate combustion gas kinetics analysis developed as an undocumented extension of techniques developed for hydrogen-fueled hypersonic ramjets (Reference 4). Other elements required to complete the analysis were developed specifically for this model.

Physical parameters input to the model include: ambient air properties (temperature, pressure, humidity, flight speed); fuel properties (temperature, hydrogen content, heating value); engine cycle parameters (ram air temperature, bypass ratio, main combustor fuel-air ratio and CO and NO_x emissions indices, fan discharge temperature, exhaust jet diameter); and exhaust jet survey probe data at several points (radial location, impact pressure, gas sample analysis). The gas analysis requires volumetric determination of CO and CO_2 with the sample dried to saturation at 32°F, and total HC (as single-carbon-atom molecules) and total NO_x with sample water content as-sampled.

The composition and properties of the gas flow at each probe measurement point are derived from the measurements. In this derivation, the gas composition at a point in the flow is assumed to be heterogeneous or time-varying. This assumption was chosen in preference to a homogeneous gas with nonequilibrium chemical composition, because examination of chemical reaction kinetics indicated that, under high temperature afterburner conditions, the reactions would proceed to equilibrium in a few inches. The existence of nonequilibrium concentrations of CO and HC in the gas samples is better explained by postulating that, over a part of the sampling time, the probe was immersed in a gas mixture much richer or much leaner than the mean fuel concentration of the sample.

The flow is assumed to be axisymmetric, so that the gas properties at each sample point are applied to the flow in an annular streamtube. The time-average properties of the flow in each tube are also computed.

The intermixing of the annular streamtubes and the diffusion of the jet air into the static or moving ambient are computed by the JETMIX analysis (Reference 2), which performs a numerical solution of the differential equations of continuity of mass, momentum, energy, gas species, and turbulence kinetic energy. The flow initially in each of the streamtubes assigned to each sample point is treated as a separate gas, as is the ambient air. These gases are treated as homogeneous, inert, ideal gases with constant specific heats. Any increase in total temperature due to chemical reactions is calculated later, but it is assumed that the density change is not sufficient to influence the mixing. In an afterburner, the combustion reactions should be essentially complete upstream of the exhaust nozzle, so that the reactions continuing into the exhaust plume are only consuming trace contaminants.

A differential equation governing the formation, convection, and decay of a characteristic of the gas heterogeneity is solved numerically throughout the flow field, using the previously calculated properties of the flow. This concept was taken from the works of Spalding (References 5, 6 and 7).

The nonreacting flow field solution provided by the JETMIX analysis and the Spalding heterogeneity solution are not used directly by the plume model, but rather are used to guide the mixing between and homogenization within a fixed number of computation tubes extended from the original streamtubes assigned to each sample point. This approach is taken to minimize the number of chemical kinetics calculations. The computation proceeds in axial steps through the plume. The gas composition in each tube is modified by intertube mixing and intratube homogenization at each step. After several steps, the gas composition is further modified by chemical reactions over a time increment corresponding to the step length and local velocity.

A choice of two chemical reaction kinetics analyses is provided. One of these, GCKP, consists of a numerical solution of simultaneous differential equations formed from the forward and reverse reaction rates of a system of 23 two- and three-molecule collision reactions. Although having a high degree of rigor, GCKP is computationally cumbersome. The alternate kinetics analysis, SCKP, utilizes approximations applicable to high-temperature combustion reactions to achieve computational efficiency. These approximations are associated with a radical pool in pseudo-equilibrium, the decay of which is paced by selected three-body reactions.

Neither chemical kinetics analysis treats consumption of unburned HC. This is done with a phenomenological technique using an empirical relation of ignition delay for kerosene as a function of temperature.

At preselected axial locations, the model performs a summation of fuel and contaminant flow in each computation tube, providing predicted values of local and overall emissions indices. When all local gas temperatures have been reduced to the point where chemical reactions cease, the emissions indices will not change with length. These are then the true residual emissions.

SECTION 2.0

COMPUTER PROGRAM DESCRIPTION

The analytical model is coded in FORTRAN IV (sometimes called FORTRAN A) language for General Electric's Honeywell 6000-series computer. The program comprises 102 subprograms, in addition to system library subroutines, and uses 166 blocks of labelled common storage. Overlay processing is used to reduce core memory requirements; the program currently requires 41K of memory.

A brief statement of the function of each subprogram follows in Section 2.1, with emphasis on the interfaces with other subprograms. The comments are intended to relate the subprogram logic to the description of analytical methods presented in Reference 1, and to aid interpretation of the program source lists in Section 4.0. The subprograms are listed in alphabetical order by deck label. Entry names may be associated with deck labels with the aid of the cross-reference chart, Table 1. The link names refer to the program overlay structure in Section 2.3 and show the general location of the subprogram in the logical structure.

Variables stored in labelled common are identified and described in detail in Section 2.2. It is presumed that the variables used internally in the various subprograms can be adequately identified through examination of the source listings in Section 4.0. The location of each common block in the overlay structure is also indicated.

Table 1. Cross-Reference List of Subprogram Entry Names (SYMDEFS).

SYMDEF	DECK	SYMDEF	DECK	SYMDEF	DECK
......	MAINP	JETINP	JETINP	REDPRE	REDPRE
//	BLKRDA	JETPRF	JETPRF	RH2	RH2
//	MAINBD	JETPRP	JETPRP	RINP	KINPP
//	CTABPR	JMESHM	JMESHM	SCALE	SCALE
//	BLOCKT	JTCTRL	JTCTRL	SCKP	SCKP
//	NAMBLK	JTEDGE	JTEDGE	SEARCH	SERCH
//	BLKTH	JTFILE	JTFILE	SETM.	MLPKG
//	KEBLOK	JTINIT	JTINIT	SETUP	SETUP
//	BLKST	JTOUT1	JTOUT1	SOLV3	SOLV3
//	BLCK	JTOUTP	JTOUTS	SPALD	SPALD
ADJMC	ADJMC	JTOUTS	JTOUTS	STPROP	STPROP
AITER1	AITER	JTSTEP	JTSTP	SUMCPD	SUMCPD
AITER2	AITER	KINET	KINET	SUMUP	SUMP
ALFA2	ALFA2	KINP	KINPP	TABPRT	TABPRT
AUTO	AUTOO	LCFIT	LCFIT	TDSEQ	TDSEQ
CAROL	CAROL	LESV	LESVV	TFMH	TFMH
CASG	CASMM	LFIT	LFIT	TFMH1	TFMH1
CASI	CASMM	LFIT1	LFIT1	THRM	THRMM
CASM	CASMM	LSPFIT	LSPFIT	THRUST	THRUST
CHEMB	CHEMB	MAIN	MAINJ	TRIDIA	TRIDIA
CINP	CUBSS	MAIN	CFILE	XSIZE	XSIZE
COCO2B	COCO2B	MFMAIN	MFMAIN	YOF	YOFX11
COEFF	COEFF	MOVE	MLPKG		
CUBS	CUBSS	MPRINT	PRINT		
DERIV	DERIV	MSHCUT	MSHCUT		
DERV	DERVV	MXFLT0	MXFLTI		
DFEQ	DFEQ	MXFLUT	MXFLUT		
EQGASH	EQGASH	MXKINO	MXKINO		
EQGAST	EQGAST	NEWFLO	NWPSI		
EQKIN	EQKIN	NEWNET	NWNET		
ERROR1	ERROR1	NEWPSI	NWPSI		
ERROR1	MAINJ	NOX2B	NOX2B		
ERRORC	ERRORC	OUT1	OUTPP		
FILL	FILL	OUT2	OUTPP		
FMPYC	MLPKG	OUT3	OUTPP		
FRGAST	EQGAST	OUTP	OUTPP		
FUEL	FUEL	PADD	PADD		
GAMCP	GAMCP	PARD	PARDD		
GAMH	GAMH	PBOLIC	PBOLIC		
GCKP	GCKP	PERR	PERRR		
GCKP1	GCKP1	PRAT	PRAT		
GMODFY	GMOD	PRED	PREDD		
HCEQ2	HCEQ3	PRED1	PREDD		
HCEQ2	HCEQ2	PROPJ	PROPJ		
HYCARB	HYCARB	PSE02	PSEQ2		
INTC	INTEE	QIREM	QIREM-		
INTE	INTEE	RATCON	RATCON		
INTG	INTG	READIN	READIN		
INTG	INTEE	READT	READT		
INTI	INTEE	READT	READT2		

2.1 The Subroutines and What They Do

Subprogram descriptions are listed velow in alphabetical order by deck label. Corresponding source listings are provided in Section 4.0. Table I may be used to locate the subprogram containing a specific entry name (SYMDEF).

Subprogram ADJMC

Entry Name: ADJMC

Type: Subroutine

Link: EMIS

Function: ADJMC is called from subroutine MXKINO to initilize the raw fuel molecule weight and adjust the polynominal coefficient A_6 to an absolute enthalpy base. These procedures are described in Appendix A (Part 3) of Reference 1.

Subprogram AITER

Entry Names: AITER1, AITER2

Type: Subroutine

Link: CPLINK

Function: Not used in PLUMOD. Call is equated to ERROR1. No list.

Subprogram ALFA2

Entry Name: ALFA2

Type: Subroutine

Link: LMXF

Function: Evaluates each array element α_{ji}, which is the fraction of flow in "NEWNET" tube j at the previous axial station that is transferred by intertube mixing to tube i at the new station. The α's are stored in common block/CMXFLT/, which also contains flows of "JETMIX" dummy gases in each tube. Fuel flows are taken from block/CNWNET/, and total flows from /CMASS/. The single argument is the number of fuel-bearing "NEWNET" tubes.

Subprogram AUTOO

Entry Name: AUTO

Type: Subroutine

Link: LGCK

Function: "Automatic elimination from error considerations of species with non-representative errors." A NASA GCKP subroutine (Reference 3) that is not used in PLUMOD. Accesses common blocks /COND/,/PQRE/, and /SKIP/.

Subprogram BLCK

Type: Block data

Link: EMIS

Function: The primary function of BLCK is to initialize the species names and molecular weights which will be used by both the GCKP and SCKP kinetics routines. Sixteen constituents are utilized in GCKP (A, NH_3, raw fuel considered inert), whereas only the first thirteen are utilized in SCKP. The constituent order is: H, O, H_2, O_2, OH, H_2O, CO, CO_2, N_2, A, NO, N, HO_2, NO_2, NH_3, C_{10} H_{10n} (or raw fuel). The names and molecular weights are stored in labeled COMMON /SNMW/. Note that the molecular weight of the fuel is dependent on the hydrogen/carbon ratio and is initialized by subroutine ADJMC.

Subprogram BLKRDA

Entry Name: none

Type: Block Data

Link: LGCK

Function: Enters specific code number, reaction rate expression coefficients, and third body efficiencies for 23 reactions, and species excluded from integration error consideration into common blocks /LSLRSV/, /REAC/, /RRAT/, and /SKIP/. In the NASA GCKP program (Reference 3), this information is read as input data.

Subprogram BLKST

Entry Name: none

Type: Block Data

Link: LGCK

Function: Enters stoichiometric coefficients for 16 species and 23 reactions in common block /SPEC/.

Subprogram BLKTH

Type: Block data

Link: PREJET, EMIS

Function: The primary function of BLKTH is to initialize the polynominal coefficients utilized by the thermodynamic routine THRM. The polynominal representation of the JANAF thermodynamic properties is discussed in Appendix A-3 of Reference 1. The data are stored in COMMON /TCOF/.

Subprogram BLOCKT

Type: Block data

Link: JETMIX

Function: JETMIX block data for turbulence parameters (Common/PROPJT/

Subprogram CAROL

Entry Name: CAROL

Type: Subroutine

Link: PREJET

Function: Computes gas sample fuel-air ratio and emissions indices from "CAROL" gas analyzer readings. The arguments, all single word variables are:

N	(independent)	H/C atom ratio of fuel
WAR	(independent)	Specific humidity of combustion air
RHC RCO RCO2 RNOX	(independent)	Gas analyzer readings for HC, CO, CO_2, and NO_x, mole fraction, with standard sample moisture content.
FAR	(dependent)	Fuel-air ratio
EIHC EICO EINOX	(dependent)	Emission indices for HC, CO, and NO_x, lb/klb fuel

The sample hydrogen content is estimated by reference to function RH2.

Subprogram CASMM

Entry Name: CASM, CASI, CASG

Type: Subroutine

Link: LGCK

Function: "Chooses the initial step (restart) formula (entry CASI) or the general step formula (entry (CASG). Sets up the augmented matrix, and computes increments." This is a modified version of the NASA GCKP subroutine of the same name (Reference 3). The modifications permit a limited number of attempts to recover from negative species concentrations by the simple expedient of repeating the step with half the step size. Calls subroutine LESV, and accesses common blocks /OPTS/, /COND/, /DERN/, and /MATX/.

Subprogram CFILE

Entry Name: MAIN

Type: Subroutine

Link: CFILE

Function: Subprogram MAIN is the main subroutine of overlay CFILE. This routine is used for two purposes:

1) Merge the output files from PREJET (file code 1), JETMIX (file code 2), and SPALDG (file code 3) to a single output file (file code 4).

2) Decollate the input file (file code 7) and restore the PREJET, JETMIX, SPALDG data on file codes 1, 2, 3 respectively.

Subprogram CHEMB

Entry Name: CHEMB

Type: Subroutine

Link: EMIS

Function: Not used. Equated to system library routine ERROR. No list.

Subprogram COCO2B

Entry Name: COCO2B

Type: Subroutine

Link: LSCK

Function: Subroutine COCO2B calculates the finite rate consumption of CO over a specified time step. The arguments to the subroutine are the static pressure and the magnitude of the time step.

The procedures for treating the CO consumption are described in Section 4.6.2 of Reference 1. Essentially, the procedure consists of a closed form integration of the rate expression for the reaction CO+OH=CO2+H. If the result indicates a CO concentration below that at mixture equilibrium, the CO mole fraction is set to the equilibrated value.

Subprogram COEFF

Entry Name: COEFF

Type: Subroutine

Link: SPALDG

Function: The function of subroutine COEFF is to evaluate the coefficients in the partial differential equation for mean square fuel concentration fluctuation intensity (G). The coefficients are calculated at the cross-stream mesh points for each axial station in the step-by-step finite difference solution procedure. Local aerodynamic properties, from subroutine STPROP, are used in the calculations. The variable K in the calling sequence for COEFF is an indicator which, in this program, is held constant (K=1).

Subprogram CTABPR

Type: Block Data

Link: Main

Function: CTABR sets the data in common /CTABPR/ to 0. (Used by subroutine TABPRT)

Subprogram CUBSS

Entry Name: CUBS, CINP

Type: Subroutine

Link: LGCK

Function: From entry CUBS, "computes coefficients for cubic spline interpolation and differentiation from input table." From CINP, performs one of the several interpolations and differentiations, using these coefficients, depending on value of ITPSZ in common block /AFUN/. This subroutine is taken intact from the NASA GCKP program (Reference 3).

Subprogram DERIV

Entry Name: DERIV

Type: Subroutine

Link: CPLINK

Function: Subroutine DERIV is used to calculate a partial derivative using divided differences that is compatible with the DFEQ routine. The arguments to the routine are the independent variable, the dependent variable, the calculated variable derivative, and the index limits in the independent variable table. This subroutine is called from JTSTEP to compute the partial derivative of the turbulent kinetic energy with respect to ψ.

Subprogram DERVV

Entry Name: DERV

Type: Subroutine

Link: LGCK

Function: "Computes all derivatives with respect to the independent variable" (time, in the PLUMOD application). This subroutine is taken intact from the NASA GCKP program (Reference 3). It accesses common blocks /OPTS/,/COND/, /SPEC/,/GHSC/,/DERN/,/NECC/, and /SABS/, and includes a call to PARD.

Subprogram DFEQ

Entry Name: DFEQ

Type: Subroutine

Link: CPLINK

Function: Subroutine DFEQ solves a partial differential equation of the general "diffusion" equation form using the methods described in Section 4.3.10 of Reference 1. The general "diffusion" equation form is:

$$\varepsilon \frac{\partial F}{\partial X} = \alpha \frac{\partial}{\partial y} (\beta \frac{\partial F}{\partial y}) + \eta \frac{\partial F}{\partial y} + \delta F + \gamma$$

The coefficients of the differential equation are passed to the routine via COMMON/PARAM/ and COMMONPARAM1/. These commons also contain the dependent and independent variables and the boundary conditions applicable for the given partial differential equation. The first of the four arguments to the routine denotes the difference form to be used for replacement of $\frac{\partial}{\partial y} (\beta \frac{\partial F}{\partial y})$.

The next two arguments are the lower and upper boundary condition indicators. The fourth argument is an error indicator switch. The reader is referred to the program listing for the various settings of the first three arguments.

The solution is effected by replacing the partial derivative by divided differences to obtain a set of linear simultaneous equations. The tri-diagonal system is solved using subroutine TDSEQ. The dependent variable and its derivative are returned in FM1 and DFM1 located in COMMON/PARAM/.

Subprogram EQGASH

Entry Name: EQGASH

Type: Subroutine

Link: PREJET

Function: Computes thermostatic properties of combustion gas mixtures at a specified enthalpy, pressure, and fuel-air ratio. Calls EQGAST with successive estimations of temperature, using QIREM to direct convergence. Arguments are the same as EQGAST, except that T is a dependent variable and H is independent.

Subprogram EQGAST

Entry Name: EQGAST, FRGAST

Type: Subroutine

Link: PREJET

Function: Computes thermostatic properties of combustion gas mixtures at a specified temperature and pressure. From entry EQGAST, the equilibrium mixture composition at the specified fuel-air ratio is computed by a call to HCEQ2. From entry FRGAST, the mixture is specified. Arguments for entry EQGAST are:

FAR	Mass ratio of fuel to dry air
WAR	Specific humidity of air
HC	H/C atom ratio of fuel
T	Temperature (°R)
P	Pressure (psia)
FIXCO } FIXNO	If .TRUE., CO or NO concentration is predetermined and excluded from equilibrium calculations

The preceeding seven arguments are independent variables.

Z Mixture composition, moles/lb mix, eleven species in order H, O, H_2, O_2, OH, H_2O, CO, CO_2, N_2, A, NO

If FIXCO = .TRUE., Z_{co} and Z_{co2} must be specified.

If FIXNO = .TRUE., Z_{no} and A_{n2} must be specified

The following six arguments are independent variables:

H	Mix enthalpy (Btu/lb)
MWT	Mix molecular weight (lb/mole)
S	Entropy (Btu/lb °R)
SPV	Specific volume (ft^3/lb)
A	Frozen sonic velocity (ft/sec)
CP	Frozen specific heat (Btu/lb °R)

The arguments FAR, WAR, HC, FIXCO, and FISNG are not used for entry FRGAST. Specie thermo properties are transmitted from subroutine THRM thru block/GHSC/.

Subprogram EQKIN

Entry Name: EQKIN

Type: Subroutine

Link: LSCK

Function: Subroutine EQKIN is called from KINET to determine the equilibrium combustion efficiency (β) and temperature (Teq.) which will be approached by the rate expressions. The equilibrium temperature is established by iteration using subroutines HXEQ2 and QIREM. Upon convergence, the equilibrium β is computed using the known fuel air ratio and the equilibrium molecular weight generated in subroutine HCEQ2.

When P DUM (20) is COMMON/CPRING) is input as 39., the kinetics scratch file defined in COMMON/DQIREM/ is changed to 39. When this situation occurs, the equilibrium temperature iteration history is recorded on the scratch file (in addition to the initial kinetics information on file 40) (see - subroutine MXKINO).

Subprogram ERROR1

Entry Name: ERROR1

Type: Subroutine

Link: PREJET & LSCK

Function: Writes contents of block/CQIREM/, containing control info for QIREM, on file 6. Transfers up to 3000 132-character BCD records (without carriage control symbols) from file number contained in single word of /DQIRM/ to file 6. Calls ERROR to generate logic trace and terminate execution.

Subprogram ERRORC

Entry Name: ERRORC

Type: Function

Link: LGCK

Function: "Computes the relative error in an integration step of size H. Determines the controlling variable." This is a renamed version of the NASA GCKP routine ERRORR. Renaming was necessary to avoid conflict with a Honeywell system library subroutine called elsewhere in PLUMOD in the event of a computation error. Common blocks /COND/,/SINT/,/PQRE/, and /SKIP/ are accessed.

Subprogram FILL

Entry Name: FILL

Type: Subroutine

Link: INLINK

Function: This routine will fill vacancies in input lists by linearly interpolating between known values. Arrays X and Y are input along with limits NA and NB on the known values.

Vacancies in the Y list are detected by the presence of a dummy value which the list was initialized to. Any such positions are filled by interpolating between the closest known values.

Subprogram FUEL

Entry Name: FUEL

Type: Subroutine

Link: PREJET

Function: Computes estimated thermodynamic properties of raw fuel vapor. Four arguments are: Fuel H/C atom ratio, temperature (°R) (both independent), enthalpy (Btu/lb), and specific heat (Btu/lb °R).

Subprogram GAMCP

Entry Name: GAMCP

Type: Subroutine

Link: JETMIX

Function: GAMCP computes the ratio of specific heats (γ) and the constant pressure heat capacity (C_p) at each transverse mesh point. The subroutine arguments are the temperature, γ, C_p, R, and the mesh point indices. For $800° R \leq T \leq 3600° R$, γ is computed as a function of temperature. Below $800° R$, $\gamma = 1.4$ and above $3600° R$, $\gamma = 1.254$.

Subprogram GAMH

Entry Name: GAMH

Type: Function

Link: JETMIX

Function: Function GAMH is called by subroutine JETINP to compute a single value of the specific heat ratio. The relations used in GAMH are identical to those used in GAMCP.

Subprogram GCKP

Entry Name: GCKP

Type: Subroutine

Link: LGCK

Function: Main driver routine for GCKP calculations. Performs certain initialization, then calls GCKP1 to perform chemical kinetics calculations on the hot part of a gas mixture. By-passed kinetics calculations if $T \leq 1500$ R. Tube number is only argument. Has access to common blocks /TROUBL/,/INDATA/,/JETDAT/,/GASCMP/,/GASTMW/,/STCTRL/,/CRPRINT/,/COND/, and /PRIN/.

Subprogram GCKPI

Entry Name: GCKPI

Type: Subroutine

Link: LGCK

Function: This is the main program of the NASA GCKP program (Reference 3) converted to subroutine form. Changes have been made to cause printing to be suppressed except just prior to a PLUMOD print station. Additional printout can be obtained by setting PDUM(11)≠0. Chemical kinetics calculations are performed by calls to CASG, INTC, INTG, INTI, KINP, OUT1, OUT2, OUT3, PRED, PRED1, and ERRORC. A non-standard return from GCKP1 occurs if a computation malfunction is encountered. Common blocks /COND/,/SINT/,/PQRE/,/PRIN/, /STCTRL/, and /CPRINT/ are accessed.

Subprogram GMOD

Entry Name: GMODFY

Type: Subroutine

Link: GMOD

Function: Reads the PREJET data from file 1, modifies the initial profiles of Spalding heterogeneity to force rapid homogenization of tubes with a mass fraction hot gas less than a specified minimum (input thru Namelist /A/), optionally modifies the mass fraction hot gas assigned to the ambient air, then restores the altered PREJET data to file 1. No arguments. Uses common block /CBITS/.

Subprogram HCEQ2

Entry Name: HCEQ2

Type: Subroutine

Link: LSCK

Function: Computes the equilibrium composition of gaseous products of combustion of CH_n fuel and moist air as a specified temperature and pressure. The arguments are the weight ratio of fuel to dry air, the specific humidity, they hydrogen-to-carbon atom ratio of the fuel, the temperature (° R), the pressure (psia), two logical variables specifying whether CO and NO are to be excluded from equilibration, and the mole fractions of species H, O, H_2, O_2, OH, H_2O, CO, CO_2, N_2, A, and NO in the mixture. If the CO concentration is to be fixed, both CO and CO_2 concentrations must be predetermined. If NO is to be fixed, both NO and N_2 concentrations must be specified. HCEQ2 uses common block /GHSC/ to transmit species thermo properties from THRMM.

Subprogram HCEQ3

Entry Name: HCEQ2

Type: Subroutine

Link: PREJET

Function: Same as HCEQ2, except that gas composition is returned as moles/pound mixture instead of mole fractions.

Subprogram HYCARB

Entry Name: HYCARB

Type: Subroutine

Link: EMIS

Function: HYCARB calculates the consumption of unburned hydrocarbon ($C_{10}H_{10n}$) contaminants in a specified reaction tube using an empirical relation of ignition delay for kerosene as a function of temperature (Section 4.6.3 of Reference 1). The hydrocarbon incipience at the upstream and current station are stored as HCINCP in COMMON/GASCMP/. When the temperature upstream is greater than 400° R, the ignition delay is calculated using the above described relation. A weighted hydrocarbon incipience including the raw hydrocarbons poured in from the lean streamtubes is computed at the initial station. The residence time/local ignition delay is then added to yield a new hydrocarbon incipience. When the accumulated incipience equals or exceeds 1, the fuel is burned instantaneously to CO and H_2. The composition of the raw fuel is set to 0, and the compositions of O_2, H_2, CO are adjusted according to the stoichiometry of the combustion reaction. The single argument to the subroutine is the streamtube index.

Subprogram INTEE

Entry Name: INTE, INTI, INTC, INTG

Type: Subroutine

Link: LGCK

Function: This is a slightly modified version of the NASA GCKP (Reference 3) subroutine of the same name. It sets up and calls for integration of the system of differential equations, then selects a step size for the next integration. The modification is a call to ERRORC rather than ERROR, which is the name of a Honeywell library subroutine used elsewhere in PLUMOD. The integration is performed by calls to AUTO, CASG, CASI, PERR, PRED, ERRORC, and SEARCH. Common blocks/OPTS/, /COND/, /SINT/, /PQRE/, and /SPEC/ are accessed.

Subprogram INTG

Entry Name: INTG

Type: Subroutine

Link: JETMIX

Function: INTG performs integration of a given function using the trapezoidal rule. The integral is returned as the third argument of the calling sequence. The integrand and independent variable tables are supplied as the first two arguments. The final two arguments are the index limits in the independent variable table.

Subprogram JETINP

Entry Name: JETINP

Type: Subroutine

Link: INLINK

Function: JETINP is the input routine which reads card input (NAMELIST $A) from file 5, calculates initial conditions at the jet discharge plane, and sets up initial profiles to start the numerical calculation. Initially, if TAPIN is T, the jet discharge parameters and the plume initial profiles generated in the PREJET overlay are read from file 1 (subroutine JTFILE). Following file input, JETINP reads card input from file 5 using NAMELIST $A. After counting the number of axial print stations, the ambient parameters and jet parameters not input are calculated. Static temperature, total pressure, and jet velocity, Mach number, and turbulence energy for the jet and external environment are then available.

If profiles are not input from either PREJET or a previous JETMIX run, approximate initial station profiles of velocity, temperature, turbulence intensity, and species mole fraction are established. In the case of a restart run, the JETMIX profiles at the restart station are reclaimed by calling JTFILE. Subroutine JETPRP is then called to calculate the stream function ψ at the initial station and initialize the data in common region /PROP J/. Finally, subroutine JT or T1 is called to print the initial station data and if TAPOT is T, JTFILE is called to store the initial profiles on file 3.

Subprogram JETPRF

Entry Name: JETPRF

Type: Subroutine

Link: PREJET

Function: The function of subroutine JETPRF is to convert nozzle exit plane survey data and engine cycle conditions to input for the JETMIX and SPALDG programs. Test data is transmitted from routines READIN and MXFLTI to JETPRF, where initial conditions and physical parameters are determined for the multi-tube, non-reacting jet flow field analysis. Certain input information can be supplied to this subroutine through the NAMELIST $A.

Subprogram JETPRP

Entry Name: JETPRP

Type: Subroutine

Link: JETMIX

Function: JETPRO is utilized to compute selected jet mixing parameters at a given print station. On the initial call, subroutines SCALE, PROPJ, and GAMCP are called to calculate the initial width of the mixing zone and values of the turbulence length scale, turbulent viscosity, specific heat ratio, and constant pressure heat capacity. The stream function coordinate ψ is then calculated by integration of equation 4.3-31 of Reference 1. For a confined mixing situation, the ψ for the streamline coincident with the outer duct wall is also determined.

After calculation of the jet properties along $\psi = 0$, the sonic line is located (if the jet is supersonic) and the entrained mass flow is determined. For a confined mixing case, the subroutine THRUST is called to determine the integrated momentum of the mixing and external streams. In this case, also, a check is made to determine if the flow is choked at the given station. If the flow is choked, and the available downstream area is less than or equal to that at the current station, the run is terminated.

Subprogram JMESHM

Entry Name: JMESHM

Type: Subroutine

Link: CPLINK

Function: Subroutine JMESHM is used to redistribute the calculation mesh in the transition/similar region of the jet. The streamlines are redistributed such that the ratio of ψ's for any two adjacent intervals is a constant. The following relation is used:

$$\psi_i = \frac{\Delta\psi_1 \, (K^{i-1}-1)}{(K-1)}$$

where: K = mesh constant

$\Delta\psi_1 = (\psi_2 - \psi_1)$

and $\psi_i/\psi_e = 1$ for the last mesh point.

Subprogram JTCTRL

Entry Name: JTCTRL

Type: Subroutine

Link: CPLINK

Function: JTCTRL is the main control routine in the calculation/output overlay. Subroutine JTSTEP is called to carry out the integration of the differential equations to the current calculation station. Upon return from JTSTEP, tests are initiated to determine if the potential core or supersonic core has disappeared or a merge point has been detected (coannular/coplanar jet). If so, the current X value is inserted as a calculation station. The subroutine JETPRP is called to calculate jet properties, and if printout was requested, JTOUTP is called to print station profiles. When the potential core disappears, JMESHM is called to redistribute the calculation mesh.

Subprogram JTEDGE

Entry Name: JTEDGE

Type: Subroutine

Link: CPLINK

Function: The prime functions of JTEDGE are to locate the outer edge of the jet mixing zone and to set indicators for addition of another streamline on the next axial solution step. Normally, the outer edge of the jet is taken as the point where the difference between the local velocity and the centerline velocity is 98 percent of the maximum velocity difference across the jet. When the variable SCLD in COMMON/SCALED/ is T, the edge is taken as the point where the ambient air mole fraction is 0.98.

After determination of the edge condition, the Y coordinates corresponding to the ψ coordinates are calculated by integration of the streamfunction relation using subroutine INTG. Finally, the indicator to add a streamline is set T if the difference between the edge velocity and the velocity at the adjacent inner mesh point is greater than 0.005. The first argument to the subroutine is the current X location. The Y and ψ values of the edge are returned as the next two arguments. The streamline addition indicator is the fourth argument in the calling sequence.

Subprogram JTFILE

Entry Name: JTFILE

Type: Subroutine

Link: JETMIX

Function: JTFILE is used to store/retrieve program data on/from tape or magnetic disc files. The subroutine has four sections which serve the following functions:

1. Read input profiles and parameters defining the jet from file 1 or file 2.
2. Write input and centerline parameters on file 2. Reclaim calculated profiles from file 3 and write them on file 2.
3. Read calculated profiles for a designated station.
4. Write calculated profiles on file 2.

Subprogram JTINIT

Entry Name: JTINIT

Type: Subroutine

Link: INLINK

Function: Initialize labeled common variables to standard settings prior to input.

Subprogram JTOUT1

Entry Name: JTOUT1

Type: Subroutine

Link: INLINK

Function: JTOUT1 prints the initial conditions at the discharge plane of the jet. Included are the parameters defining the jet such as T, P, U, etc. and the initial profiles of normal coordinate, streamfunction, velocity, temperature, turbulence intensity, and species mole fractions. For the confined mixing situation, the stream function value at the outer duct surface is also printed.

Subprogram JTOUTS

Entry Name: JTOUTS, JTOUTP

Type: Subroutine

Link: CPLINK

Function: Subroutine JTOUTS (ENTRY JTOUTP) is the primary output routine for the JETMIX program. JTOUTS prints a summary print of the overall station paramaters of the jet. Included in this dimensionless output are the X station location, the width of the mixing zone the ψ coordinate of the jet edge, the centerline velocity, temperature, turbulence intensity, total pressure ratio, and total temperature ratio, the location of the sonic line, and the relative entrainment ratio. For a confined mixing case, this output is followed by a summary of the dimensional parameters related to the ducted mixing problem. This output consists of the X location, the duct and centerbody coordinates, the dimensionless duct coordinate and iterated value, the pressure, flow state (subsonic, supersonic, choked), the integrated momentum (thrust) and the Mach number, velocity, and static temperature in the external unentrained inviscid stream.

Entry JTOUTP also produces a print of the dimensionless and dimensional profiles at a given calculation station. The output includes the dimensionless ψ coordinate, the stream function, the Mach number, turbulence intensity, and both the dimensionless and dimensional velocities, temperatures and total pressures. The profiles of species mole fractions are printed on a following page if a species diffusion case is being run. If an output file is requested, the profiles are temporarily saved on file 3.

Subprogram JTSTEP

Entry Name: JTSTEP

Type: Subroutine

Link: CPLINK

Function: Subroutine JTSTEP is the main calculation routine for integration of the partial differential equations described in Section 4.3.9 of Reference 1. The numerical procedures for this process are given in Section 4.3.10. The differential equations are solved sequentially in the following order:

1. X - Momentum

2. Turbulent kinetic energy

3. Species continuity (if applicable)

4. Energy

The most current values of each dependent variable are used as each equation is solved using subroutine DFEQ. At each intermediate step between calculation stations, routine JTEDGE and XSIZE are called to locate the outer edge of the jet and adjust the current X stepsize. If the potential or supersonic core has disappeared, or a merge station detected (JTEDGE), a premature return is made to subroutine JTCTRL to insert the pertinent station. Otherwise, the integration continues to the next calculation station. Additional routines called at intermediate steps are SCALE, PROPJ, and MSHCUT. The latter subroutine is called to redistribute the calculation grid only if the maximum allowable number of mesh points is exceeded.

Subprogram KEBLOK

Type: BLOCK DATA

Link: LSCK

Function: The primary function of KEBLOK is to initialize parameters utilized by the SCKP kinetics routines. Specific items preset are the rate constant Arrehenius coefficients, activation energies, and 3-rd body catalytic efficiencies for the pertinent chemical reactions described in Section 4.6.2 of Reference 1.

Subprogram KINET

Entry Name: KINET

Type: Subroutine

Link: LSCK

Function: Subroutine KINET is utilized to compute the kinetic history of the reaction streamtube over a specified time step. The calling arguments to the subroutine are the static pressure, the static enthalpy, the fuel/air ratio (lbm/lbm), and the residence time of the step. The defining equations and the calculation procedure for advancing the kinetics solution is given in Section 4.6.2 of Reference 1. The static pressure, enthalpy, and fuel air ratio are considered constant over the step.

On the initial entry, the local combustion efficiency (B) and density are established at the beginning of the time step. On the second call, subroutine EQKIN is entered to establish the equilibrium conditions which the chemical reactions will approach. The kinetics solution is then carried out in a sequence of small steps (NTSTEP in COMMON/CKINET/) using the following procedure:

1. The consumption of CO is computed by a call to COCO2B which uses the relations defined in Section 4.6.2 of Reference 1.

2. The rate expression for the 3-body reactions is integrated in closed form to determine a new combustion efficiency (B defined in terms of molecular weight).

3. The static temperature and composition of the non-equilibrium mixture is then established by iteration using subroutines PSEQ2 and QIREM.

4. Finally, the consumption of nitrogen oxide contaminants (NOX) is calculated using subroutine NOX2B.

5. The procedure is repeated until the time interval specified by the fourth argument is reached.

Subprogram KINPP

Entry Name: KINP, RINP

Type: Subroutine

Link: LGCK

Function: This is a modified version of the NASA GCKP (Reference 3) subroutine of the same name. It performs initialization and reads input data. The modifications delete most of the input, as the chemical reaction system is built into labelled common in PLUMOD, rather than being input as data, and the initial composition is determined internally from previous calculations. Namelist SPROB is read for integration controls on the first call to GCKP only. Subprograms called include CINP, CUBS, and THRM. There are no arguments. Common blocks/ LTUS/, /OPTS/, /COND/, /REAC/, /RRAT/, /AFUN/, /SPEC/, /SINT/, /TCOF/, /PRIN/, /XVSA/, /SNMW/, /KOUT/, /GHSC/, /PQRE/, /SKIP/, /NECC/, /MISC/, /INDX/, /LSLRSV/, /CINSAV/, and /STCTRL/ are accessed.

Subprogram LCFIT

Entry Name: LCFIT

Type: Subroutine

Link: JETMIX

Function: LCFIT is a least squares curve fitting routine utilized in the confined mixing option. After fitting a least squares parabola to the i th and i + 1 points (the X of interest being between X (I) and X (I+1)) this routine will return an interpolated value for the function, the derivative of the function at the point, or the integral of the function from one specified X to another.

Input to the routine consists of X and Y arrays for the points on the curve, the number of points, a list of X values at which the calculations are to be done, the constant of integration (if applicable), the number of calculation points, indicator for type of calculation, and an array of coefficients of the parabola if available from a previous call.

Subprogram LESVV

Entry Name: LESV

Type: Subroutine

Link: LGCK

Function: "This routine is a general double precision linear equation solver. In this program (GCKP) it is used to compute the increments (of the dependent variables)." It is taken intact from the NASA GCKP program (Reference 3). Common blocks /COND/ and /MATX/ are accessed.

Subprogram LFIT

Entry Name: LFIT

Type: Subroutine

Link: LMXF

Function: LFIT is used to provide linear interpolation, integration, or differentiation of a function specified in tabular form. Input to the routine consists of X and Y arrays for the points on the curve, the number of points, a list of X values at which the calculations are to be done, the constant of integration (if applicable), the number of calculation points, and an indicator for the type of calculation (0 = interpolation, -1 = integration, 1 = differentiation).

Subprogram LFITL

Entry Name: LFIT1

Type: Subroutine

Link: SPALDG

Function: Same as LFIT

Subprogram LSPFIT

Entry Name: LSPFIT

Type: Subroutine

Link: LMSF

Function: Calls LFIT, and has the same arguments.

Subprogram MAINBD

Type: Block Data

Link: Main

Function: Initialize the data in the commons /TROUBL/ and /CBITS/.

Subprogram MAINJ

Entry Name: MAIN, ERROR 1

Type: Subroutine

Link: JETMIX

Function: Subprogram MAIN is the main program of overlay JETMIX. Its prime function is to load in the subordinate input and calculation overlays and call the routines JETINP, JTCTRL, and JTOUTS for input, calculation, and output respectively.

Entry point ERROR1. Sets ERR (block/TROUBL/) to T to indicate a JETMIX error and returns to the main program in link o.

Subprogram MAINP

Entry Name: ......

Type: Main Program

Link: Main

Function: The main program MAINP provides control for execution of the programs in the PLUMOD system. The prime functions of this routine are as follows:

1. Read the initial three case identification cards.

2. Read and print the fixed format program cards, load the proper program overlay, and call the main subroutine of the overlay.

3. Print error messages and/or the "end of job" conditions.

Depending on the "program" card TAPIN, TAPOT settings, files 1, 2, 3, 4, or 7 are rewound prior to entry of the main subroutine of the called overlay. Upon return, end of file marks are written on the generated output files.

When an error condition (ERR = T) is returned by JETMIX or SPALDG, the program CFILE is called to merge the partial output files such that the job can be restarted.

Subprogram MFMAIN

Entry Name: MFMAIN

Type: Subroutine

Link: LMXF

Function: MFMAIN provides control for the calculation of the mixing/homogenization of the heterogeneous gas streams. On the initial entry, NAMELIST $INPUT is read and mixing step parameters are initialized. Subroutine READT is then called to read JETMIX and SPALDG data from files 2 and 3 and initialize aero-data regions. A maximum of five data stations may be retained in memory at one time.

The solution is advanced stepwise in the axial direction by consecutively calling NEWNET to generate reaction tubes and MXFLUT to perform the mixing/homogenization process. The number of mixing steps before return for a kinetics calculation is specified by NIST in COMMON/STCTRL/. Return also occurs when an overall print station is reached. In this instance, FINAL in COMMON/STCTRL/ is set T to direct control to the output overlay. If the current X location goes beyond the midpoint of the data stations in memory, the JETMIX and G profiles are shifted upstream and READT is recalled to read new station information. Stepwise printout of the mixing steps occurs if PDUM (18) in COMMON/CPRINT/ is non-zero.

Subprogram MLPKG

Entry Names: FMPYC, MOVE, SETM

Type: Subroutine

Link: MAIN

Function: A GMAP subroutine drawn from the AEG Scientific Library to perform high-speed manipulation of arrays in core memory. Cannot be coded in FORTRAN because of variable number of arguments. No list included.

FMPYC is used for floating-point multiplication of a constant times the elements of an array and storage of the resultant products.

The first argument is the number of arrays to be multiplied by the constant. The second is the constant to be used. Then follows for each array, the array to be multiplied, the array in which to store the products, and the number of elements in the array to be multiplied.

MOVE moves data from one array to another or shifts data within the confines of the same array.

The first argument in the calling sequence is the number of arrays to be moved. There are then four arguments for each array to be moved. The first of these is the initial location of the array to be moved. The second is the destination of the array to be moved. The next is the number of elements of the array to be moved. The fourth is the storage increment between successive elements of the array.

If data are to be shifted upward within an array, the number of elements should be negative.

SETM sets regions of memory to a specified value.

The first argument is the number of arrays to be set to a given value. The second is the data word to be used. Then follows for each array the beginning of the array and the number of elements of the array to be set to the specified value.

Subprogram MSHCUT

Entry Name: MSHCUT

Type: Subroutine

Link: CPLINK

Function: MSHCUT is used to redistribute the calculation mesh if the number of points exceeds the maximum (NMAX - COMMON/INPJET/). The arguments to the routine are the region indicator, the stream-function, and the number of profile points. The number of mesh points is reduced by NRED (COMMON/UMESH/).

For the potential core region, a new semi-uniform mesh is calculated. In the transition/similar region, a new array of stream function coordinates is produced such that the ratio of $\Delta\psi$'s for any two adjacent intervals is a constant.

Subprogram MXFLTI

Entry Name: MXFLTO

Type: Subroutine

Link: PREJET

Function: Establishes the detailed properties of the two-part model of the hetergeneous gas flow from the gas analysis and impact pressure measurements made with a probe at one point in the engine exhaust plane. The probe point number is the single variable in the calling sequence; the remaining input data are taken from common block /INDATA/. Calculated properties are stored in blocks/JETDAT/ and /GASCMP/. Scratch file 40 is used to store iteration histories for subsequent dumping by subroutine ERROR1 in the event of convergence failure. Utilizes subroutines RH2, FUEL, PRAT, CAROL, EQGASH, EQGAST, FRGAST, and several utility subroutines.

Subprogram MXFLUT

Entry Name: MXFLUT

Type: Subroutine

Link: LMXF

Function: Performs the intertube mixing and intratube homogenization over an axial step to define flow, fuel concentration, enthalpy, and mixture composition of the hot and cold parts of each "NEWNET" tube. Uses ALFAZ to define the intertube mixing coefficients. Calls HYCARB to compute HC consumption in the step. Has no arguments, but uses common blocks /CREACT/, /CMXFLT/, /CSPECI/, /CNWNET/, /CCHECK/, /JETDAT/, /GASCMP/, /GASTMW/, /CMASS/, /CPRINT/, /STCTRL/, /CAXIAL/, and /CAGAIN/.

Subprogram MXKINO

Entry Name: MXKINO

Type: Main Program

Link: EMIS

Function: MXKINO provides control for computation of the mixing/homogenization of the heterogeneous gas streams, the subsequent chemical kinetic calculations in the hot (fuel rich) portion of each reaction tube, and finally the computation and output of the plume emission indices at specified streamwise measuring stations.

Initially, the fuel molecular weight and adjusted thermo coefficient for the raw fuel are established by a call to ADJMC, data regions are initialized, and control input NAMELIST/A/ is read.

The computation for each streamtube then proceeds downstream by successively performing mixing/homogenization steps (link LMXE) and chemical kinetic steps (links LGCK, LSCK). After each mixing step, the molecular weights are computed and the static temperatures are established by a call to subroutine TFMH. The chemical kinetics step is accomplished either by use of the NASA-GCKP program (Section 4.6.1 of Reference 1) or the GE-SCKP program (Section 4.6.2). Prior to initiation of a chemical kinetics step, scratch file 40 is rewound and initial data are saved in the event of a calculation malfunction. When the calculation has reached a streamwise measuring station, as given by the PRINT variable in COMMON/OPCTRL/, link LFIN is entered to provide computation and display of the residual emission indices. The calculation is terminated when the final measuring station is reached.

Program errors are detected if the variable ERR in COMMON/TROUBL/ is set T. When this situation occurs, the last generated kinetic history is printed by a call to subroutine ERROR1.

Subprogram NAMBLK

Entry Name: None

Type: Block Data

Link: LGCK

Function: Enters alphanumeric data in common blocks /LTUS/,/OPTS/,/SPEC/, and /KOUT/. In the NASA GCKP program (Reference 3), these data are used as input.

Subprogram NOX2B

Entry Name: NOX2B

Type: Subroutine

Link: LSCK

Function: Subroutine NOX2B calculates the rate of formation of NO by integration of the rate expressions described in Section 4.6.2 of Reference 1. The single rate equation, developed by assuming pseudo equilibrium concentrations of N_2, O_2, OH, O, H, and "steady state" concentration of N is integrated numerically using a Runge-Kutta technique.

Subprogram NWNET

Entry Name: NEWNET

Type: Subroutine

Link: LMXF

Function: Defines boundaries of "NEWNET" tubes, which are tubes of constant fuel flow, by interpolation in the "JETMIX" flow field solution arrays. Computes flows of "JETMIX" dummy gases in each tube, and mean velocity, heterogeneity and reaction time. Has no arguments, but uses common blocks /CAGAIN/, /CINPJT/,/JETDAT/,/CCHECK/,/CNWNET/,/CSPECI/,/CINPUT/,/CAXIAL/,/CLOCAL/, /CMXFLT/,/CMASS/,/GASTMW/, and /CRRINT/.

Subprogram NWPSI

Entry Name: NWPSI, NEWFLO

Type: Subroutine

Link: LMXF

Function: Establish a set of values of normalized stream function, ranging from 0.0 to 1.0, so that each "NEWNET" tube will contain approximately the same number of points. The 50-element array containing the stream functions is the first argument of the NWPSI calling sequence; the second is the number of stream functions in the set. The flows in the "NEWNET" tubes are taken from common block /CMASS/. At the initial computation station, these flows are established through entry NEWFLO, which has no arguments, using radii, velocities, and specific volumes from block /JETDAT/.

Subprogram OUTPP

Entry Name: OUTP, OUT1, OUT2, OUT3

Type: Subroutine

Link: LGCK

Function: This is a modified version of the NASA GCKP (Reference 3) subroutine of the same name. The modifications were made for a GE-AEG stand-alone version of GCKP, and have no effect in the PLUMOD application. OUTPP performs unit conversions and prints the various pages of output. Output data are obtained from common blocks /OPTS/,/COND/,/SINT/,/KOUT/,/REAC/,/RRAT/,/AFUN/,/SPEC/, /XVSA/,/NECC/,/PQRE/,/DERN/,/SKIP/,/GHSC/, and /SABS/.

Subprogram PADD

Entry Name: PADD

Type: Subroutine

Link: CPLINK

Function: PADD is used to add streamlines to the calculation mesh at the edge of the jet. This routine is called from JTSTEP if the JTEDGE subroutine detected that points were to be added at the next station due to entrainment of the ambient stream.

In the potential core region, one point is added at the same spacing as the adjacent two outer points. In the transition/similar region, a new point is added based on the constant ratio rule for $\Delta\psi$'s of adjacent intervals.

Subprogram PARDD

Entry Name: PARD

Type: Subroutine

Link: LGCK

Function: "Computes all mixed partial derivatives." This subroutine is taken intact from the NASA GCKP program (Reference 3). It accesses common blocks /OPTS/,/COND/,/SPEC/,/REAC/,/RRAT/,/GHSC/,/NECC/,/SABS/, and /DERN/.

Subprogram PBOLIC

Entry Name: PBOLIC

Type: Subroutine

Link: SPALDG

Function: The function of subroutine PBOLIC is to set up the system of simultaneous finite difference equation (for the cross-stream mesh points at an axial station) in the step-by-step solution of the parabolic partial differential equation for mean square fuel concentration fluctuation intensity (G). A generalized Crank-Nicolson finite difference representation is used, allowing either explicit or implicit formulation of the equations. However, the solution procedure, in subroutine TRIDIA, assumes an implicit formulation (simultaneous equation) in either case, so that there is no advantage to be gained by utilizing the capability for explicit formulation. A tridiagonal system of simultaneous linear algebraic equations is established, utilizing coefficients calculated in subroutine COEFF. The resulting set of equations is then solved in subroutine TRIDIA. The variable K in the calling sequence for PBOLIC is an indicator which, in this program, is held constant (K = 1).

Subprogram PERRR

Entry Name: PERR

Type: Function

Link: LGCK

Function: This is a slightly modified version of the NASA GCKP (Reference 3) subprogram of the same name. It predicts the computation error that can be expected from using given numerical integration step size. The modification is a call to ERRORC rather ERROR, which is the name of a Honeywell library routine used elsehwere in PLUMOD. Common blocks /COND/,/SINT/,/DERN/, and /PQRE/ are accessed.

Subprogram PRAT

Entry Name: PRAT

Type: Function

Link: PREJET, LFIN

Function: Calculates ratio of probe impact pressure to local static pressure for a perfect gas of specified Mach number (subsonic or supersonic) and specific heat ratio, which are the two arguments.

Subprogram PREDD

Entry Name: PRED, PRED1

Type: Subroutine

Link: LGCK

Function: This subprogram is taken directly from the NASA GCKP program (Reference 3). It "performs all necessary pre-derivative calculations." Common blocks /OPTS/,/COND/,/SPEC/,/REAC/,/RRAT/,/GHSC/,/NECC;, and /SABS/ are accessed.

Subprogram PRINT

Entry Name: MPRINT

Type: Subroutine

Link: LMXF

Function: PRINT is the output routine for the mixing/homogenization program, called from MFMAIN when PDUM(1B) [COMMON/CRRINT/] is non-zero. Items printed are the current axial location, and the radius, fuel flow, total mass flow, velocity, mass fraction of rich flow, fuel/gas ratios (rich and lean), Spalding G function, and residence time for each reaction streamtube.

Subprogram: PROPJ

Entry Name: PROPJ

Type: Subroutine

Link: JETMIX

Function: PROPJ computes the properties of a laminar or turbulent jet. The parameters determined are the density, dynamic viscosity, turbulence length scale, eddy viscosity, diffusion parameter and k_{eff}/C_p. The latter four properties are used for calculation of the turbulent mixing flow field. No computed quantities are returned through the call sequence.

Subprogram: PSEQ2

Entry Name: PSEQ2

Type: Subroutine

Link: LSCK

Function: Computes the psuedo-equilibrium composition of gaseous products of combustion of CH_n fuel and moist air at a specified temperature and degree of dissociation. The arguments are the weight ratio of fuel to dry air, the specific humidity, the hydrogen-to-carbon atom ratio of the fuel, the temperature (° R), the double precision reaction efficiency β (Reference 1, Appendix A-2), two logical variables specifying whether CO and NO are to be excluded from pseudo-equilibrium, and the mole fractions of species, H, O, H_2, OH, H_2O, CO, CO_2, N_2, A, and NO in the mixture. If the CO concentration is to be fixed, both CO and CO_2 concentrations must be predetermined. If NO is to be fixed, both NO and N_2 concentrations must be specified. PSEQ2 uses common block /GHSC/ to transmit species thermo properties from THRM.

Subprogram: QIREM

Entry Name: QIREM

Type: Subroutine

Link: PREJET, LSCK

Function: QIREM is an iteration routine for determining the roots of functions with maxima or minima using a quadratic interpolation for the root evaluation.

With each call to this routine, an independent coordinate X and its corresponding functional value Y are supplied. Based on the last three such points, a new trial X at which the desired Y is estimated to occur is calculated.

Input is supplied to the routine through a calling sequence

```
CALL QIREM (X, Y, YJP, QV)
```

and a labeled common

```
COMMON /CQIREM/ YTOL, YO, DYDX, CTRMAX
```

The input variables are:

X = trial value of x.

Y = y calculated for the current trial value of x.

XJP = x-jump to be taken before the desired y has been spanned if DYDX = 0. The sign is for a positive error; that is, the x-jump is XJP if (Y-YO)>0 and the x-jump is -XJP if (Y-YO)<0. XJP also indicates the branch of the curve for which the solution is to be found, if two solutions exist. XJP will be positive for the branch with negative slope, as in the sketch on page 1; XJP will be negative for a branch with positive slope.

QV(1)= counter which must be set to zero before the first entry.

YTOL = convergence tolerance on (Y-YO).

YO = given ordinate, y_o, for which the solution y=f(x) is desired.

DYDX = optional estimate of the curve slope. A non-zero value should be supplied if a maximum point of a solution near a maximum point is being sought.

CTRMAX = maximum number of trials before yielding to defeat. Twenty-five trials will be allowed if CTRMAX=0.

The output from the routine is:

X = next trial value of x.

QV(1) = 0. if the solution has been found, |Y-YO|<YTOL;

QV(5) = 0. if no solution can be found but the maximum point has been found within YTOL; otherwise QV(5) is non zero.

QV, known as the quire vector, must be dimensioned by at least 8. It contains the counter, three previously tried values of x, a second counter, and the three corresponding values of y.

The input supplied via the labeled common is, to a certain extent, optional as indicated below:

1. The value of CTRMAX needs to be specified only if the limiting number of trials is not 25.

2. If DYDX and YO are not specified (i.e. DYDX=YO=0.), the QIREM routine operates exactly the same as subroutine QIRE (although, the calling sequences to QIREM and QIRE are different).

3. YTOL must be specified unless the test for convergence is performed before calling QIREM.

The parameter DYDX indicated which of two options are desired. If DYDX=0 the iteration proceeds without reference to the local calculated values of slope until a solution value or maximum point has been passed. Then the quadratic (or linear in the case of two available points) interpolation is performed to obtain subsequent x trial values. This particular option has utility when the curve has irregularities of the type illustrated below; in this case the direction of the x-jump is determined by the ordinate error. (See the definition of XJP on the previous page).

Subprogram: RATCON

Entry Name: RATCON

Type: Subroutine

Link: LSCK

Function: Subroutine RATCON is utilized to compute the rate constants for the 3-body recombination reaction scheme described in Section 4.6.2 of Reference 1. The pressure is input to the routine via the calling sequence. The temperature is obtained as the third variable in COMMON/PSEQ/.

In the process of evaluating the rate constants, the effects of the hydroperoxyl radical are included using a "steady state" assumption for the HO_2. The rate constant for the reaction $H+O_2+M=HO_2+M$ is then adjusted using a sequence of four 2-body shuffle reactions involving the HO_2 radical. All forward rate constants are calculated as a function of temperature using Auehenius type relations of the form:

$$k_f = AT^n e^{-(\Delta E_A/RT)}$$

Reverse rate constants for the pertinent 2-body reactions are expressed in terms of the forward rate constants using the equilibrium constants calculated from the Gibb's Free Energy (Subroutine THRMM).

Subprogram: READIN

Entry Name: READIN

Type: Subroutine

Link: PREJET

Function: Initializes, then reads engine test data and cycle parameters from file 05 via namelist TSTDAT. Sets data to standard values if not input. Sorts and stores data in common blocks /INDATA/, /JETDAT/, /GASCMP/, and /OPCTRL/. Calls MXFLTO to initialize heterogeous flow model at each sample location. Calculates ambient air properties using EQGAST.

Subprogram: READT

Entry Name: READT

Type: Subroutine

Link: SPALDG

Function: The function of subroutine READT is to read PREJET and JETMIX output files to obtain aerodynamic properties used in the calculation of mean square fuel concentration fluctuation intensity (G). The radial profile of G at the nozzle exit plane and mean fuel concentrations in the constituent gases are obtained from the PREJET output file. Physical constants describing the jet, as well as radial profiles of aerodynamic properties at several axial stations in the jet, are obtained from the JETMIX output files. Radial profiles of mean fuel concentration are calculated (from constituent gas concentration profiles) in READT and stored along with the profiles of axial velocity, turbulent kinetic energy, density and turbulence length scale. The profiles are retained in "current" storage, in groups of five stations at a time. As the calculation of G moves downstream, profiles at upstream stations are removed from current storage and replaced with profiles from stations further downstream. The variable JENTRY in the calling sequence indicates whether or not READT has been called previously:

IENTRY = 0	first call to READT
IENTRY > 0	READT has been called previously

Subprogram: READT2

Entry Name: READT

Type: Subroutine

Link: LMXF

Function: READT is called from MFMAIN to read JETMIX and SPALDG station profiles for use in the mixing calculation. On the initial call, subroutine REDPRE and NEWFLO are called to reestablish the nozzle discharge conditions and preset the flow rates in the reaction tubes. Initially, five sets of profiles are read, including ψ, Y, U, T, G, Mach number, and species mole fractions. The outer edge of the jet is established and a new set of profiles are interpolated using a uniform grid with a maximum of 50 points. After the radii and ψ are redimensionalized (Section 4.5.1 of Reference 1), the profiles are moved to the MXFLUT common region /CINPUT/. On successvie entries, only a single set of station profiles are read (see - MFMAIN).

Subprogram: REDPRE

Entry Name: REDPRE

Type: Subroutine

Link: LMXF

Function: Subroutine REDPRE is used to restore the commons defining the jet discharge plane condtions and the jet properties generated in the PREJET overlay. The data are stored in COMMONS/INDATA/, /JETDAT/, /GASCMP/, and /OPCTRL/. Before return, the temperature of the rich gas fraction of each streamtube is calculated by a call to TFMH1 and stored in COMMON/GASTMW/.

Subprogram: RH2

Entry Name: RH2

Type: Function

Link: PREJET

Function: Estimates mole fraction H_2 based on the measured mole fraction CO in the gas sample, which is the single argument of the function.

Subprogram: SCALE

Entry Name: SCALE

Type: Subroutine

Link: JETMIX

Function: The prime function of subroutine SCALE is to determine the boundaries of the mixing zone for calculation of the turbulence scale in subroutine PROPJ. For the single free jet, the width of the mixing zone is normally determined using the jet edge (subroutine JTEDGE) and the point where the axial velocity deviates from a constant value (potential core). When the variable SCLD in common /SCALED/ is T, the inner edge is determined as the point where the n-th constituent concentration has reached the value ALXLIM by turbulent diffusion. This latter parameter is also stored in COMMON /SCALED/. For a coannular/coplanar jet, the variable MERGE is checked to determine whether the solution is upstream or downstream of the merge station. If upstream, the coordinates at the edges of both the primary and secondary jets are set as well as the widths of the mixing zones to the two jets. If

Subprogram: SCALE (Continued)

downstream of the merge point, the boundaries of the mixing zones are determined using linear curve fits for the nozzle-merge station lines.

The arguments to the subroutine are the flow-field variable (U, α) to be used for determination denoting a single or coannular/coplanar jet, the axial region (core, transition, similar), and the current X station. No computed quantities are returned through the call sequence.

Subprogram: SCKP

Entry Name: SCKP

Type: Main Program

Link: LSCK

Function: Subroutine SCKP is the main control program for the high temperature Special Chemical Kinetics Procedure for calculation of the chemical reactions in a given streamtube. On entry, the initial temperature of the rich fraction of the streamtube is obtained from COMMON/GASTMW/. If less than 1500° R, the reactions are assumed quenched and control is returned to the calling program. If greater than 1500° R, subroutine THRM is called to compute the dimensionless enthalpy of the individual species. The constituents N, HO_2, NO_2, NH_3, and C_nH_{2n} are not evaluated directly in the kinetics procedure. To assure conservation of energy, these species are removed by splitting the mixture into an active fraction and an inert fraction. After adjusting the composition and the static enthalpy of the active fraction, subroutine KINET is called to compute the kinetics step over the time inverval specified by variable TAU in COMMON/GASTMW/. Upon completion of the step, the static temperature and composition are moved into the upstream storage locations and return is made to the calling program. The single argument to the subroutine denotes the streamtube index.

Subprogram: SERCH

Entry Name: SEARCH

Type: Subroutine

Link: LGCK

Function: "This routine uses an optimal sequential search technique to find X in the interval (A,B) such that F(X) = FOFX." H is taken intact from the NASA GCKP program (Reference 3). F, FOFx, A, X, and B are arguments. Common block /SINT/ is accessed.

Subrpogram: SETUP

Entry Name: SETUP

Type: Subroutine

Link: SPALDG

Function: The function of subroutine SETUP is to set up JETMIX output profiles of aerodynamic properties for interpolation along streamlines in the calculation of mean square fuel concentration fluctution intensity (G). A cross-stream linear interpolation is performed, at each axial station for which JETMIX output is available, to establish the local aerodynamic properties on the streamlines used in the finite difference solution for G. The property matrices are then transposed to facilitate interpolation along the individual streamlines, so that local properties can be determined at intermediate axial stations. The variable IRST in the calling sequence indicates whether or not the JETMIX output profiles in current storage include the initial profile (X=0):

IRST = 0	initial station is in current storage
IRST > 0	initial station is not in current storage

The variable K is an indicator which, in this program, is held constant (K = 1).

Subprogram: SOLV3

Entry Name: SOLV3

Type: Subroutine

Link: PREJET, LSCK

Function: Solves a set of three simultaneous linear algebraic equations:

$$\sum_{j=1}^{3} A_{ij}X_j = B_i \qquad i = 1,2,3$$

The vectors A, B, and X are the first three arguments of the calling sequence (X is the solution vector). The fourth argument is a logical variable which is returned .TRUE. if no solution exits.

Subprogram: SPALD

Entry Name: SPALD

Type: Subroutine

Link: SPALDG

Function: SPALD is the main program in link SPALDG, for the calculation of mean square fuel concentration fluctuation intensity (G) in the jet. SPALD controls the step-by-step finite difference procedure used to solve the parabolic partial differential equation for G, and performs all output functions. Certain input information (constants) can be supplied to this program through the namelist /CHANGE/.

Subprogram: STPROP

Entry Name: STPROP

Type: Subroutine

Link: SPALDG

Function: The function of subroutine STPROP is to calculate local aerodynamic properties, for the cross-stream mesh points, in the step-by-step solution procedure used in determining the distribution on mean square fuel concentration fluctuation intensity (G) in the jet. The properties are determined by linear interpolation along streamlines between the axial station for which JETMIX output profiles are available. STPROP uses the transposed property matrices, created by subroutine SETUP, for interpolation. The variable K in the calling sequence is an indicator which, in this program, is held constant (K = 1).

Subprogram: SUMCPD

Entry Name: SUMCPD

Type: Function

Link: CPLINK

Function: Subroutine SUMCPD is called from JTSTEP to calculate the diffusion flux term used in the solution of the energy equation. The single argument of the function is the local mesh point index.

Subprogram: SUMP

Entry Name: SUMUP

Type: Subroutine

Link: LFIN

Function: At specified axial stations, computes the mean properties of the flow in each NEWNET tube, and synthesizes the predicted values of gas sample analysis. Integrates the fuel and contaminant flows over all tubes and computes overall emissions indices. Prints a table of calculated data, and punches profile information on cards suitable for subsequent plotting or other postprocessing by the user. Uses routines TFMH1 and PRAT, and derives data from common blocks /GASCMP/, /OPCTRL/, /JETDAT/, /COAXIAL/, /CMASS/, /INDATA/, /STCTRL/, and /CBITS/.

Subprogram: TABPRT

Entry Name: TABPRT

Type: Subroutine

Link: MAIN

Function: TABPRT is used for labeled diagnostic printout of tabular data. The first argument in the calling sequence is the name of the array to be printed. The second is the location of the array. The third is the number of elements and the last is the number of columns to be used for the print format. Through the common region CTABPR is also input the location of the first element of the array to be printed (1st, 2nd, ect.).

The structure of the elements of the array are checked against various criteria to determine whether they are floating point numbers, integers, BCD, or octal. The formats for the printout are then set accordingly.

Subprogram TDSEQ

Entry Name: TDSEQ

Type: Subroutine

Link: CPLINK

Function: TDSEQ is used to solve a system of tri-diagonal linear equations. The coefficients of the system are passed to the routine via the first argument of the CALL sequence. The solution is returned as the first column of the array. The last three arguments are the number of equations, the maximum column size, and an error indicator.

Subprogram TFMH

Entry Name: TFMH

Type: Subroutine

Link: EMIS

Function: Subroutine TFMH is utilized to compute the static temperature and constant pressure heat capacity of the rich and lean fractions of a given stream tube. The subroutine is logically identical to TFMH1. However, species compositions and enthalpies are picked up from COMMON/GASCMP/ and calculated temperatures and heat capacities are stored in COMMON/GASTMW/. The single argument specifies the upper value of the reaction tube index. (See description of COMMON/GASCMP/.

Subprogram TFMH1

Entry Name: TFMH1

Type: Subroutine

Link: EMIS

Function: TFMH1 is used to determine the static temperature and the constant pressure heat capacity given a mixture composition and static enthalpy. The temperature is determined by iteration using subroutine THRM to calculate a trial enthalpy for comparison with the input enthalpy. If the iteration does not converge to a tolerance of 0.01° R within 30 trials, an error message is printed and current values are returned. The arguments to TFMH1 are the concentration (moles i/lbm mixture), the static enthalpy (BTU/lbm), the calculated temperature and heat capacity. Single values of the latter two parameters are returned via the call sequence.

Subprogram THRMM

Entry Name: THRM

Type: Subroutine

Link: PREJET, EMIS

Function: Computes dimensionless thermodynamic properties of up to 25 chemical species at the specified temperature, using polynomial coefficients from common block /TCOF/. The temperature (Kelvins) is the first argument of the calling sequence. If the second argument has value 0.0, only enthalpy is computed. The number of species is taken from block /COND/, which also contains logical variable NEXT which is set .TRUE. if the temperature is too far out of the range of the polynomial coefficients. If the temperature is within limits, the dimensionless enthalpy, free energy, entropy, specific heat, and derivative of specific heat with respect to temperature are computed and stored in block /CHSC/.

Subprogram THRUST

Entry Name: THRUST

Type: Subroutine

Link: JETMIX

Function: For the confined mixing case, subroutine THRUST computes the integrated momentum (thrust) at a given station. If the mixing zone has intersected a plug or center-body, the program is terminated.

Subprogram TRIDIA

Entry Name: TRIDIA

Type: Subroutine

Link: SPALDG

Function: The function of subroutine TRIDIA is to solve the tridiagonal system of simultaneous linear algebraic equations formed by implicit finite difference formulation of the partial differential equation for mean square fuel concentration fluctuation intensity (G). A set of simultaneous equations is formed for the cross-stream mesh points at each axial station in the step-by-step solution procedure. The variable J in the calling sequence is an indicator which, in this program, is held constant (J = 2), while the variable NN is equal to the current number of streamlines minus one.

Subprogram XSIZE

Entry Name: XSIZE

Type: Subroutine

Link: CPLINK

Function: XSIZE controls the integration step size in the streamwise (axial) direction. In the potential core region, the step-size is proportional to the width of the mixing region with the magnitude set by the value of the input variable CXPC. In the transition and similar regions, DX is proportional to the radius of the jet, with a magnitude dependent on input CXTP.

The arguments to the routine are the calculated step size, the current X location, the reference scale, the pertinent region, and a terminal step indicator (LAST). LAST is set T if the step size would go beyond a calculation station. In this case, the returned step size is reduced to identically hit the calculation station.

Subprogram YOFXII

Entry Name: YOF

Type: Function

Link: JETMIX, LSCK

Function: YOFX performs a linear interpolation in a specified table of X, Y values. The additional call sequence parameters are the desired value of X[1] and the index limits in the X table.

The I index in the common YOFXI is an estimate of the interval that X lies in (between X (I-1) and X(I)) and the IA and IB indexes are limits on the interval from which points can be used for the interpolation.

The routine will extrapolate linearly if the specified X is outside the range indicated.

2.2 Variables in Common Storage

Descriptions of the contents of blocks of labelled common follow, in alphabetical order, by block name. Within each block, variables are listed according to position occupied in the block. In some cases, the variable name may differ between routines and a typical name is given. Pertinent dimension and type information are included with the variable name (R = Real, I = Integer, L = Logical). Variables normally containing BCD data are typed as H = Hollerith, even though they may have real or integer names.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
ACONVG (JETMIX)					Confined mixing data storage
	1-100	YCD	R	100	Jet edge at each calculation station, in.
	101-200	PD	R	100	Static pressure at each calculation station, psia.
	201-300	INDC	H	100	Flow indicator SUBSON SUPSON CHOKED
	301	CHOKE	L	1	Choke indicator T - Flow choked
	302	CHOKED	L	1	Auxiliary choke indicator T - denotes previous choked station.
ADAM01 (MAIN)					Identification block
	1-10	NAME	H	10	User name
	11-20	ADDRES	H	10	User address
	21-30	TITLE	H	10	Case title
	31-40	IDENT	M	10	Not used
ADAM02 (MAIN)					
	1	ENDJOB	L	1	T = case has terminated normally
	2-4	DUMMY	R	3	Not used
BC (JETMIX)					Current boundary conditions at edge of jet
	1	UEDGE	R	1	u/u_J at edge of jet (velocity)
	2	EEDGE	R	1	e/e_J at edge of jet (turbulent kinetic energy)
	3	THEDGE	R	1	T/T_J at edge of jet (temperature)
BCO (JETMIX)					Coannular jet - Primary jet boundary conditions
	1	UO	R	1	u/u_J in core of secondary jet (velocity)
	2	EO	R	1	e/e_J in core of secondary jet (turbulent kinetic energy)
	3	THO	R	1	T/T_J in core of secondary jet (temperature)

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
BCMIX2 (JETMIX)					Confined mixing - communication
	1	GRADU			Not Used
	2	TW			
	3	MUW			
	4	RHOW			
	5	PTE	R	1	Inviscid region stagnation pressure, psia
	6	TTE	R	1	Inviscid region stagnation temperature, °R
BCMOL (JETMIX)					Boundary conditions for species diffusion equations
	1-12	ALEDGE	R	12	Mole fractions at edge of jet
	13-24	ALO	R	12	Mole fractions in core of secondary jet
CADCNT (SPALDG)					
	1	MAX	I	1	Maximum number of equally spaced streamlines (mesh points) for finite difference calculation
	2	NS1	I	1	Number of JETMIX streamlines at initial axial station (x=0)
	3	NXS	I	1	Number of JETMIX output stations currently in use for determination of aerodynamic property profiles (interpolation)
	4	NSZMAX	I	1	Maximum number of equally spaced streamlines required for JETMIX output stations in current usage
	5	NSZ1	I	1	Number of equally spaced streamlines at initial axial station (x=0)
CAGAIN (EMIS)					Used by NEWNET
	1	AGAIN	I	1	Set = -1 after first call to avoid reinitialization
CARRY (JETMIX)					JETMIX new case indication
	1	NEW	L	1	T = new case - causes initial values to be set by JTINIT

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CAXIAL (EMIS)					Axial distances in jet diameters (from block CINPJT)
	1	X	R	1	Present computation station
	2	XL	R	1	Next pring station
	3	ALOGX	R	1	ℓn (x+1)
CBITS (MAIN)					Utility common
	1	BITS	R	1	GE junk work 037777777777
	2	BLANK	H	1	Hollerith blank 6hbbbbbb
CBNDRY (SPALDG)					
	1	CX	R	1	Current axial step-size ($\Delta x/d_p$) for finite difference calculation of "G"
	2	DXPRN	R	1	Axial distance ($\Delta x/d_p$) between stations at which "G" profiles are printed
	3	JT	I	1	Indicator (in this program, (JT ≡ 1)
CBODY (JETMIX)					Confined mixing - communication
	1-100	YCB	R	100	Coordinates of centerbody, in.
	1010200	CLSPCB	R	100	Coefficients for curve fit of centerbody
	201	YCB1	R	1	Local normal coordinate on centerbody
	202	UCL1	R	1	Local velcoity (u/u_j) on centerbody
CCHECK (EMIS)					Continuity check parameters used in NEWNET and MXFLUT
	1-12	SPECIF	R	12	Total flow of each dummy gas at current station (diagnostic only)
	13-24	WKTOT	R	12	Total flow of each JETMIX dummy gas in initial jet.
	25-36	RATIO	R	12	Not used

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CCOCO2 (LSCK)					Communication block between KINET and COCO2B
	1	CORATE	L	1	T = co-kinetics f = co-pseudo-equilibrium
	2	XCOI	R	1	Initial CO mole fraction
CCONST (SPALDG)					
	1	CONST1	R	1	Constant ($= \alpha_t \sqrt{g_c Je}$)
	2	CONST2	R	1	Constant ($= 12\ C_{G1}/u_p d_p$)
	3	CONST3	R	1	Constant ($= 12\ C_{G2}\ d_p/\alpha_t\ u_p$)
	4	CONST4	R	1	Constant ($= 1/\sigma_G$)
CCOUNT (SPALDG)					
	1-5	X	R	5	Axial locations (x/d_r) of JETMIX output profiles in current storage
	6-10	NSL	I	5	Number of JETMIX streamlines at JETMIX output stations in current storage
	11	NX	I	1	Number of JETMIX output stations in current storage
	12	XMAX	R	1	Axial location (x/d_r) of last available JETMIX output station
CDXSAV (EMIS)					Used by MFMAIN
	1	DXSAV	R	1	Basic axial step size in jet diameters
CENDS (SPALDG)					
	1	JSTART	I	1	Indicator (in this program, JSTART = 1)
	2	JENDS	I	1	Number of equally spaced at current axial station in finite difference calculation
CEQKIN (LSCK)					General data storage (EQKIN and KINET)
	1	BK	R	1	
	2	A1K	R	1	Not used
	3	DELHNO	R	1	
	4	DEN	R	1	Current value of $(1/m_c - 1/m_u)^{-1}$
	5	XMWEQ	R	1	Equilibrium molecular weight
	6	XNO	R	1	Current NO mole fraction
	7	XCO	R	1	Current CO mole fraction

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CFILK (LMXF, SPALDG)					Scale parameter for file records (READT, READT2)
	1	CSC	R	1	Scaling constant for file record identification
CGJET (CFILE)	200				Variables stored on various output files but not used in the program reading the file. Used also as dummy or temporary storage input/output operations.
CINIT (SPALDG)					
	1	XX	R	1	Current axial station (x/d_p) in finite difference calculation
	2	DPSI	R	1	Increment of strain function ($\Delta\psi \sim lbm/ft^3$) between adjacent equally spaced streamlines at current axial station
CINPJT (EMIS)					Properties of the initial jet extracted by READT2 from the JETMIX output files.
	1	DIAJ	R	1	Jet diameter, inches
	2	MJET		1	Not used
	3	TJET		1	Not used
	4	PTJET		1	Not used
	5	VJET	R	1	Reference velocity of jet, fps
	6	TIJET		1	Not used
	7	PE		1	Not used
	8	VE		1	Not used
	9	ME		1	Not used
	10	TIE		1	Not used
	11	TE		1	Not used
	12-61	X	R	50	Axial stations (in jet diameters) at which JETMIX placed profile data on its output files
	62	GAM		1	Not used
	63	RG		1	Not used
	64	PR		1	Not used
	65	PRT		1	Not used

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CINPUT (SPALDG)					
	1-500	PSI	R	100,5	Stream function ($\psi \sim$ lbm/ft^3)
	501-1000	R	R	100,5	Radius ($2r/d_p$)
	1001-1500	U	R	100,5	Axial velocity (u/u_p)
	1501-2000	E	R	100,5	Turbulent kinetic energy (e/e_p)
	2001-2500	RHO	R	100,5	Density ($\rho \sim$ lbm/ft^3)
	2501-3000	XLN	R	100,5	Turbulent length scale ($L_t \sim$ ft)
	3001-3500	F	R	100,5	Fuel concentration (lb_m/lb_m mixture)
					(1-3500: Profiles at the JETMIX output stations in current storage (JETMIX streamliner))
CINPUT (EMIS)					READT profile properties at five axial stations
	1-5	XX	R	5	Axial location of each station, $\ell n(x+1)$
	6-255	RJR	R	5,50	Radius squared (sq. in.)
	256-505	UJR	R	5,50	Velocity, ft/sec
	506-755	RHOJR	R	5,50	Density (not used)
	756-4005	XKJR	R	5,50, 13	Mole fraction of JETMIX dummy gas
	4006-4255	PSIJR	R5,50		Stream function, lb/sec
	4256-4505	CJR	R	5,50	Spalding heterogeneity parameter,(lb fuel/lb mix)^2
	4506-4510	NJJ	I	5	Number of profile points
	4511	NII	I	1	= 5
CINSAV (EMIS)					GCKP input storage block
	1-57	HGCKP	R	57	Common block used to temporarily save input to GCKP

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CJETOT (SPALDG)					
	1	GCJ	R	1	Constant (g_c x J = 25040 lbm-ft^2/Btu-sec^2)
	2	DIAJ	R	1	Initial jet diameter (d_r ∿ in)
	3	VJET	R	1	Mass-weighted-average initial jet axial velocity (u_p ∿ ft/sec)
	4	EJET	R	1	Initial jet turbulent kinetic-energy (e_p ∿ Btu/lbm)
	5	NXTA	I	1	Total number of available JETMIX output stations
CJMIX1 (CFILE, LMXF)	2316				Variables stored on various output files but not used in the program reading the file. Used also as dummy or temporary storage during input/output operations.
CJMIX2 (CFILE, LMXF)	2507				Variables stored on various output files but not used in the program reading the file. Used also as dummy or temporary storage during input/output operations.
CKHO2 (LSCK)					Subroutine RATCON - forward rate constants for Hydorperoxyl reactions
	1	K5	R	1	$H + O_2 + M = HO_2 + M$
	2	K6	R	1	$H + HO_2 = 2OH$
	3	K7	R	1	$OH + HO_2 = H_2O + O_2$
	4	K8	R	1	$O + HO_2 = OH + O_2$
	5	K9	R	1	$H + HO_2 = H_2 + O_2$
CKINET (LSCK)					Main communication common - SCKP/KINET/RATCON
	1	TIME	R	1	Current time into reaction stop, sec
	2	DTIME	R	1	Sub-step size for SCKP reaction, sec

Block Name (Link)	Words	Typical Variable Name	Typical Variable Type	Dim.	Description
CKINET (cont'd.)					
	3	NTSTEP	I	1	Number of intermediate sub-steps for SCKP reaction
	4	XMOLWO	R	1	Not used
	5	CRC	L	1	Not used
	6	NREAC	I	1	Number of reactions 3 body + 2 body
	7-15	RC1	R	9	Arrehnius constants for SCKP reactions
	16-24	RC2	R	9	Temperature exponents for SCKP reactions
	25-33	RC3	R	9	Activation energies for SCKP reactions, cal/gm mole
	34-88	TBE	R	11,5	3^{rd} body efficiencies for recombination reactions
	89-93	RCON	R	5	Rate constants for 3-body recombination reactions
	94	TK	R	1	Reaction temperature, °K
	95	ENTRY1	L	1	Flag denoting initial entry to KINET T = initial entry
CLOCAL (LMXF)					Profile properties of current axial station, interpolated by NEWNET from the five READT profiles
	1-50	RJX	R	50	Radius squared at each READT profile point (sq. in.)
	51-100	UJX	R	50	Velocity at each READT profile point (Ft/sec)
	101-150	DUM10		50	Not used
	151-800	XKJX	R	50,13	Mole fraction of each JETMIX dummy gas at each READT profile point
	801	NJ	I	1	Number of READT profile points
CLSPF (JETMIX, LMXF)					LCFIT, LSPFIT, LFIT Interpolation common
	1	II	I	1	Saves interpolation index for use on the next entry

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CMASS (EMIS)					Mass flows and fuel flows in each NEWNET tube
	1-12	ZMASS	R	12	Total flow in each tube at previous computation station, lbm/sec
	13-24	ZMASSH	R	12	Total flow in each tube at present station, constructed from ZMASS and ALPHA (block CMXFLT)
	25-36	FUELL	R	12	Fuel flow in cold part at present station (only FUELL(1) is used), lbm/sec
	37-48	FUELR	R	12	Fuel flow in hot part at present station (only FUELR(1) is used), lbm/sec
CMATRX (LMXF)					In MXFLUT
	1-12	INTR1		12	Not used
	13	INTR2		1	Not used
	14	DET		1	Not used
	15	IFACTR		1	Not used
CMOLES (LMXF)					Profile data read by READT2 from JETMIX output files
	1-1200	ALX	R	100, 12	Mole fraction of each dummy gas at each point in profile
CMXFLT (EMIS)					Flows of JETMIX dummy gases in NEWNET tubes, and inter-tube mixing parameters
	1-144	W		12,12	W_{ij} is flow of gas i in tube j at previous axial station (lbm/sec)
	145-300	WHAT		13,12	W_{ij} is flow of gas i in tube j at present axial station (lbm/sec) (Last i is mean fuel concentration in tube j)
	301-444	ALPHA		12,12	α_{jk} is fraction of flow in tube j at previous station that is transferred to tube k at present station

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description	
CNERR (JETMIX)					Constants and error indicator common	
	1	BITS	R	1	Junk word used for testing	
	2	ERR	L	1	Error indicator (T = error condition)	
	3	GC	R	1	(g_c) 32.174 ft lb_m/lb_f sec^2	
	4	GCJ	R	1	$(g_c J)$ (32.174*778)ft^2 lb_m/Btu sec^2	
	5	FOOT	R	1	Conversion factor 12 inches/ft	
CNFILE (MAIN)					Files present on merge/ decollation file	
	1	NF	I	1	1 = PREJET data 2 = PREJET, JETMIX data 3 = PREJET, JETMIX, SPALDG data	
CNUNET (SPALDG)						
	1-5	PSJMAX	R	5	Maximum value of stream of function ($\psi \sim lbm/ft^2$) at each JETMIX output station in current storage	
	6-10	NSLZ	I	5	Number of equally spaced streamlines at JETMIX output stations in current storage	
	11-260	RZ	R	50,5	Radius $(2r/d_p)$	Profiles at the JETMIX output stations in current storage (equally spaced streamlines)
	261-510	UZ	R	50,5	Axial velocity (u/u_p)	
	511-760	EZ	R	50,5	Turbulent kinetic energy (e/e_p)	
	761-1010	RHOZ	R	50,5	Density $(\rho \sim lbm\text{-}ft^3)$	
	1011-1260	XLNZ	R	50,5	Turbulent length scale $(L_t \sim ft)$	
	1261-1510	FZ	R	50,5	Fuel concentration (lbm/lbm mixture)	

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CNWNET (EMIS)					Profile properties at current axial station, interpolated by NEWNET from the five READT profiles
	1-50	WFCJX	R	50	Cumulative fuel flow from center of each READT profile print.
	51-100	DUM6		50	Not used
	101-112	WFCUM	R	12	Cumulative fuel flow to outer boundary of each NEWNET tube.
	113-162	DUM7		50	Not used
	163-212	PSIJX	R	50	Stream function (lb/sec) at each READT profile point
COLIMT (LSCK)					
	1	XCOLIM	R		Limiting (equilibrium) CO mole fraction - used by COCO2B
COND (PREJET, EMIS)					Properties of the reacting gas mixture, used by GCKP
	1	DVAR	R	1	The dependent variable (V=0)
	2	AREA	R	1	Flow area (not used)
	3	MDOT	R	1	Mass flow (not used)
	4	P	R	1	Pressure, atm.
	5	IVAR	R	1	The independent variable (time, sec)
	6	V	R	1	Velocity, cm/sec
	7	RHO	R	1	Density, gm/cm^2
	8	T	R	1	Temperature, K
	9-33	SIGMA	R	25	Gas composition, gmol/gm mix
	34	LS	I	1	Number of species in mixture (= 16)
	35	LSP3	I	1	LS+3
	36	NEXT	L	1	Malfunction has occurred
CONSTF (JETMIX)					Flow scaling (JETMIX)
	1	CON1	R	1	$V_J r^n$ n=1 Plane n=2 Axisymmetric

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CORED (JETMIX)					Primary jet core data.
	1	XCORE	R	1	Axial coordinate of the point where jet core disappears.
	2	CORE	L	1	T = core has disappeared.
	3	CORED	L	1	T = core disappearance station has been processed.
	4	CORSTP	L	1	T = core has disappeared, do not re-process.
COREQN (EMIS)					Rate constant selection for reaction - $CO + OH \rightleftharpoons CO_2 + H$.
	1	COREQ	L		T = use special GE rate constants. F = use standard rate constants.
CPARAM (LMXF)					Profile data read by READT2 from JETMIX output files.
	1-100	U	R	100	Velocity, ft/sec.
	101-200	T	R	100	Static temperature, °R.
	201-300	TOT	R	100	Not used.
	301-400	XMACH	R	100	Mach number.
CPBOLI (SPALDG)					
	1-50	G	R	50,1	Mean-square fuel concentration fluctuation intensity (G).
	51	ALPHA	R	1,1	α (coefficient in P.D.E. for determination of G).
	52-101	BETA	R	50,1	β (coefficient in P.D.E. for determination of G).
	102-151	GAMM	R	50,1	γ (coefficient in P.D.E. for determination of G).
	152-201	DELTA	R	50,1	δ (coefficient in P.D.E. for determination of G).

(CPBOLI words 1-201, bracketed: Profiles at the current axial station in the finite difference calculation of G (equally spaced streamlines).)

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CPCOEF (PREJET)					
	1	CA	R	1	a) Coefficients for the first NC-1 constituent (inert) gases. These are coefficients for the determination of constant pressure specific heat (c_p ∿ Btu/ lbm-mole °R') as a quadratic function of local static temperature (T ∿ °R): $c_p = a + bT + cT^2$.
	2	CB	R	1	b)
	3	CC	R	1	c)
	4	CAA	R	1	a) Coefficients for the NC^{th} gas (ambient).
	5	CBA	R	1	b)
	6	CCA	R	1	c)
CPRFL (SPALDG)					
	1-100	PSI	R	100	Stream function (ψ ∿ lbm/ft^3). Profiles at an individual JETMIX output station as it is being loaded into current storage.
	101-200	Y	R	100	Radius ($2r/d_p$).
	201-300	UD	R	100	Axial velocity (u/u_p).
	301-400	ED	R	100	Turbulent kinematic energy (e/e_p)
	401-500	RHO	R	100	Density (ρ ∿ lbm/ft^3).
	501-600	XLN	R	100	Turbulence length scale (L_t ∿ ft).
	601-1800	ALX	R	1200	Molar concentrations of constituent gases (inert species). The variable ALX actually represents a 100 x 12 matrix.
	1801	NPD	I	1	Number of JETMIX streamlines.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CPRFL (cont'd)					
	1802-1813	FSPECI	R	12	Fuel concentration in constituent (inert) gases (lbm/lbm mixture).
	1814	NC	I	1	Number of constituent (inert) gases.
	1815-1864	X	R	50	Axial locations (x/d_r) of available JETMIX output stations.
CPRINT (EMIS)					Diagnostic Print common.
	1-20	PDUM	R	20	Input switches for diagnostic output in mixing and kinetics routines.
CPRJET (CFILE)					
	1387				Variables stored on various output files but not used in the program reading the file. Used also as dummy or temporary storage during input/output operations.
CPROF (LMXF)					Profile data read by READT2 from JETMIX output files.
	1-100	PSI	R	100	Dimensionless stream function.
	101-200	Y	R	100	Dimensionless radius or radius squared.
	201-300	UD	R	100	Dimensionless velocity.
CPROGM (MAIN)					
	1	PROGM	H	1	MAINP - program name current program in execution.
CPROP (JETMIX)					Constant coefficients for mixing scale calculation - (Single jet).
	1-10	CT	R	10	Coefficients of scale calculation.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CPROP2 (JETMIX)					Coannular jet - constant coefficients for mixing scale calculation.
	1	CTP	R	1	Scale constant for primary jet.
	2	CTS	R	1	Scale constant for secondary jet.
	3	CTM	R	1	Scale constant for region downstream of merge point.
CPROPJ (LMXF)					Profile data read by READT2 from JETMIX output files (and not used).
	1-100	RHO	R	100	Density.
	101-200	XLN	R	100	Turbulence scale.
CPSEQ2 (LSCK)					
	1	SPM	R	1	
	2	SPMU	R	1	Not used.
	3	SPMC	R	1	
CQIREM (PREJET, LSCK)					Control parameters for QIREM.
	1	YTOL	R	1	Convergence tolerance on difference between YO and calculated value of dependent variable.
	2	YO	R	1	Desired value of dependent variable.
	3	DYDX	R	1	Estimate of first derivative of function.
	4	CTRMAX	R	1	Maximum number of tries before calling ERROR1.
CRATE (LSCK)					
	1	RATE	R	1	Rate of 3-body recombination reactions.
CREACT (EMIS)					In MXFLUT.
	1-12	RHOREA		12	Not used.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CREFDA (PREJET)					
	1	DIAJ	R	1	Initial jet diameter ($d_p \sim$ in).
	2	MJET	R	1	Mass-weighted-average initial jet Mach number.
	3	TIJET	R	1	Initial jet turbulence intensity (u'_p/u_p).
	4	VJFT	R	1	Mass-weighted-average initial jet axial velocity ($u_p \sim$ ft/sec).
	5-104	GJFT	R	100	Profile at mean-square fuel concentration fluctuation intensity ($G \sim lbm^2/lbm^2$) at initial axial station (γ=0).
	105	PE	R	1	External (ambient) static pressure ($P_e \sim$ psia).
	106	TE	R	1	External (ambient) static temperature ($T_e \sim$ °R).
	107	TTE	R	1	External (ambient) turbulence intensity (u'_p/u_p).
	108	VE	R	1	External (ambient) axial velocity ($u_e \sim$ fps).
	109	GEX	R	1	External (ambient) mean-square fuel concentration fluctuation intensity ($G_{ex} \sim lbm^2/lbm^2$).
	110	RG	R	1	Gas constant (ft-lbf/lbm-°R).
	111	PR	R	1	Prandtl number.
	112	PRT	R	1	"Turbulent" Prandtl number.
	113	SC	R	1	Sutherland constant for viscosity calculation ($S_c \sim$ °R).
	114	TREF	R	1	Reference temperature for viscosity calculation ($T_{ref} \sim$ °R)
	115	MUREF	R	1	Reference viscosity for viscosity calculation ($\mu_{ref} \sim$ lbm/ft-sec).

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CREFDA (cont'd.)					
	116	DIFF	L	1	T = species diffusion calculated, F = species diffusion not calculated.
	117	NC	I	1	Number of constituent gases (inert species) in jet and external flow.
	118-129	CNAME	R	12	Names of constituent gases.
	130-141	ALE	R	12	External (ambient) molar concentrations of constituent gases.
	142-153	SCM	R	12	Effective Schmidt numbers of constituent gases.
	154-189	CPC	R	36	Coefficients in polynomial representation of molar heat capacities of constituent gases (a_1, b_1, c_1, a_2, ...): $C_{pi} = a_i + b_iT + c_iT^2$.
	190	NJ	I	1	Number of JETMIX streamlines in jet at initial axial station ($\gamma = 0$)
	191	NM	I	1	Maximum number of JETMIX streamlines before mesh re-distribution occurs.
	192	CT1	R	1	C_{t1} } Constants in the empirical equation for turbulence length scale in the potential core region.
	193	CT2	R	1	C_{t2}
	194	CT3	R	1	C_{t3} } Constants in the empirical equation for turbulence length scale in the transition region.
	195	CT4	R	1	C_{t4}
	196	CT5	R	1	C_{t5}
	197	CT6	R	1	C_{t6}
	198	CT7	R	1	C_{t7}
	199	CT8	R	1	C_{t8} Constant in the empirical equation for turbulence length scale in the fully developed region.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description	
CREFDA (cont'd.)						
	200-299	Y	R	100	Radius ($2r/d_p$).	Profiles at the initial axial station (JETMIX streamlines).
	300-399	UD	R	100	Axial velocity ($u \sim$ ft/sec).	
	400-499	THD	R	100	Static temperature ($T \sim$ °R).	
	500-599	TID	R	100	Turbulence intensity (u'/u_p).	
	600-1799	ALX	R	1200	Molar concentrations of constituent gases. (Actually represents a 100 x 12 matrix)	
CSLDAT (SPALDG)						
	1-50	SY	R	50	Stream function ($\psi \sim$ lbm/ft^3).	
	51-100	RAD	R	50	Radius ($2r/d_p$).	Profiles at the current axial station in the finite difference calculation of G.
	101-150	VEL	R	50	Axial velocity (u/u_p).	
	151-200	TKE	R	50	Turbulent kimetic energy (e/e_p).	
	201-250	DEN	R	50	Density ($\rho \sim$ lbm/ft^3).	
	251-300	TLS	R	50	Turbulence length scale ($L_t \sim$ ft).	
	301-350	FAR	R	50	Fuel concentration (lbm/lbm mixture).	
CSFALD (CFILE)	51				Variables stored on various output files but not used in the program or temporary storage during input/output operations.	

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CSPARE (LMXF)					Temporary storage for redistributed JETMIX and SPALDG profile points in READT2.
	1-1600	H	R	1600	Temporary values of Y^2, U, RHO, ALX, G.
	1601-1650	PSIR	R	50	Values of ψ/ψ_c generated by NEWPSI
	1651-1700	G	R	50	Values of G, ψ, and radius read from SPALDG profile data output files.
	1701-1750	SY	R	50	
	1751-1800	RAD	R	50	
CSPECI (EMIS)					Miscellaneous parameters used by MXFLUT, NEWNET, etc.
	1	NSPECI	I	1	Number of JETMIX dummy gases (includes ambient air).
	2	NF	I	1	Storage location for fuel flow (= NSPECI + 1).
	3	DX	R	1	Current axial step size in jet diameters (from block CINPJT).
CSTART (SPALDG)					
	1-100	GJET	R	100	Profile of mean-square fuel concentration fluctuation intensity(G) at initial axial station (x=0).
	101	GEX	R	1	External (ambient) value of mean-square fuel concentration fluctuation intensity (G).
CTABPR (MAIN)					Subroutine TABPRT.
	1	ITAB	I	1	Sets array index for starting location of printing.
CTHETA (SPALDG)					
	1	THETA	R	1	Crank-Nicolson factor for finite difference solution of the parabolic partial differential equation for G (THETA = 0 $\sim$ explicit solution, THETA = 1/2 $\sim$ Crank-Nicolson, THETA = 1 $\sim$ fully implicit solution.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CTHETA (cont'd.)					
	2	II	I	1	Indicator (in this program, II ≡ 1).
CTRANS (SPALDG)					
	1-250	RZT	R	5,50	Transpose of RZ — Transposed profile matrices used for determination of profiles at intermediate axial stations by interpolation along the "equally spaced" streamlines (see CNUNET for definition of RZ, UZ, EZ, RHOZ, XLNZ and EZ).
	251-500	UZT	R	5,50	Transpose of UZ
	501-750	EZT	R	5,50	Transpose of EZ
	751-1000	RHOZT	R	5,50	Transpose of RHOZ
	1001-1250	XLNZT	R	5,50	Transpose of XLNZ
	1251-1500	FZT	R	5,50	Transpose of FZ
CTRIDI (SPALDG)					Solution of simultaneous finite difference equations for "G" profile at current axial station.
	1-150	A	R	50,3	Tridiagonal coefficient matrix
	151-200	B	R	50	"right-hand side" vector.
CTRL (JETMIX)					Calculation control.
	1	NXTA	I	1	Number of main calculation stations.
	2	CMPRS	L	1	F = incompressible. T = compressible.
	3	QJET	L	1	F = isothermal jet. T = non-isothermal jet.
	4	TURBJ	L	1	F = laminar jet. T = turbulent jet.
	5-14	COEF	R	10	Constant quantities used in the calculation of the difference equation coefficients

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CTRL (cont'd.)					
	15	NPU	I	1	Number of upstream profile points.
	16	NPD	I	1	Number of downstream profile points.
	17	DXC	R	1	Current ΔX to next station.
	18	XU	R	1	Upstream X location.
	19	XD	R	1	Downstream X location.
	20-819	DSTOR	R	800	Dummy - scratch storage.
CTRL2 (JETMIX)					Control storage - coannular jet.
	1	EDGEI	R	1	Y location of edge of primary jet.
	2	SFI	R	1	ψ (streamfunction) value at EDGEI.
	3	MERGE	L	1	F = coannular jets have not merged. T = coannular jets have merged.
	4	XMERGE	R	1	X location of merge points of coannular jets.
	5	YMERGE	R	1	Y or R location of merge point of coannular jets.
	6	SLOPEI	R	1	Slope of straight line from primary nozzle lip to merge point.
	7	SLOPEO	R	1	Slope of straight line from secondary nozzle lip to merge point.
	8	CEPTI	R	1	Intercept of straight line from primary nozzle lip to merge point.
	9	CEPTO	R	1	Intercept of straight line from secondary nozzle lip to merge point.
CXLOCA (EMIS)					Used by READT2.
	1-5	XX	R	5	Five axial distances in jet diameters for which READT2 has stored JETMIX profile data.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
CYOLD (LMXF)					In MXFLUT.
	1-12	YRICHO		12	Not used.
DFIT (JETMIX)					Confined jet - curve fit.
	1-100	CLSP	R	100	Coefficients for LSP curve fit of outer wall contour.
DICTRL (JETMIX)					Control storage - species diffusion equations.
	1	DIFF	L	1	T = solve species diffusion equations. F = Do not solve species diffusion equations.
	2-11	CND	R	10	Constant quantities used in the calculation of the diffusion difference equation coefficients.
DIFEQI (JETMIX)					Species diffusion - input common.
	1	NC	I	1	Number of species - maximum = 6.
	2-7	CNAME	H	6	Species names.
	8-13	ALJ	R	6	Primary jet - reference species mole fractions
	14-19	ALJO	R	6	Secondary jet - reference species mole fraction.
	20-25	SCM	R	6	Species Schmidt Numbers.
	26-31	TCPRF	R	6	Not used.
	32-37	HCPRF	R	6	Not used.
	38-55	CPC	R	3,6	Species molar heat capacity. Coefficients of parabolic fit vs. absolute temperature.
DPBETA (LSCK)					
	1	BETADP	D	2	Double precision value of β - used by PSEQ2

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
DQIREM (PREJET, EMIS)					Kinetics/Equilibrium T calc. scratch file.
	1	KINFIL	I	1	File code for output of SCKP initial conditions/Equilibrium T calculation iteration history to be dumped by call to ERROR1.
EDGE (JETMIX)					Coordinate of jet edge.
	1	YEDGE	R	1	Normal coordinate (Y) at jet edge.
	2	SFEDGE	R	1	Streamfunction (ψ) at jet edge.
ERASE (LSK, PREJET)	36				Scratch common used by QIREM iteration routine.
FILK (JETMIX)					Scale parameter for file records.
	1	CSC	R	1	Scaling constant for file record identification.
FLOBAL (JETMIX)					Confined jet - parameters for pressure - area iteration.
	1	MAXIT	I	1	Maximum allowable P-A iterations.
	2	SUPB	L	1	T = select supersonic branch during P-A iteration. F = select subsonic branch.
	3	PSID	R	1	Streamfunction at duct outer wall.
	4	YDD	R	1	Local duct outer wall Y.
	5	YDC	R	1	Local center body Y.
	6	P1	R	1	Upstream static pressure.
	7	P2	R	1	Downstream static pressure.
	8	UCL	R	1	Local ℄ or centerbody u/u_j.
	9	TOL	R	1	Iteration tolerance - P-A iteration.
	10	UPSTRM	L	1	T = upstream of intersection of mixing jet and duct wall. F = downstream of intersection of mixing jet and duct wall.
	11	CVG	L	1	T = P-A iteration has converged.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
GASCMP (PREJET, EMIS)					Detailed properties of hot and cold gas parts of each tube flow at previous and present computation station (see Table 2 for subscript key).
	1-24	YY	R	12,2	Mass fraction hot part of gas (subscripts K,L).
	25-72	FUEL	R	2,12,2	Fuel concentration, lb fuel/lb mix (subscripts J,K,L).
	73-120	ENTH	R	2,12,2	Absolute enthalpy, Btu/lbm (subscripts J,K,L).
	121-888	CONC	R	16,2,12,2,	Species concentrations, miles I/lb mix (subscripts I,J,K,L).
	889-912	HCINCP	R	12,2	"Reactive incipiency" of hydrocarbons in hot part (subscripts K,L).
	913-936	U	R	12,2	Velocity, ft/sec (subscripts K,L).
	937-1104	OTHER		168	Unused space.
GASTMW (EMIS)					Communication between mixing/homogenization and kinetics routines (see Table 2 for subscript key).
	1-48	TG	R	2,12,2	Static temperature, °R [TG(J,K,L)].
	49-96	MWTG	R	2,12,2	Molecular weight, lbm/lb mole [MWTG(J,K,L)].
	97-120	TAU	R	12,2	Reaction time for a [TAU (K,L)] given streamtube.
	121-168	CPG	R	2,12,2	Constant pressure heat capacity, Btu/lbm °R [CPG(J,K,L)].
GHSC (PREJET, EMIS)					Thermo properties of pure chemical species.
	1-25	GRT	R	25	Dimensionless Gibbs free energy G/RT.
	26-50	HRT	R	25	Dimensionless enthalpy H/RT.
	51-75	SR	R	25	Dimensionless entropy S/R.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
GHSC (cont'd.)					
	76-100	CPR	R	25	Dimensionless heat capacity C_p/R.
	101-125	DCPR	R	25	$(dC_p/dT)/R$.
INDATA (PREJET, EMIS)					Engine cycle parameters and nozzle exit survey probe measurements.
	1	N,HC	R	1	Fuel hydrogen/carbon atom ratio.
	2	HF	R	1	Absolute enthalpy of fuel, Btu/lbm.
	3	WAR	R	1	Water/air mass ratio in ambient air.
	4	T2	R	1	Fan inlet total temperature, °R.
	5	BETA	R	1	Engine by-pass ratio.
	6	T25	R	1	Fan discharge total temperature, °R.
	7	FAR5	R	1	Turbine discharge fuel/air mass ratio.
	8	EINO2C	R	1	NO_x emission index of main combustor, lb NO_2/lb fuel.
	9	PO	R	1	Ambient static pressure, psia.
	10-20	RCO	R	11	Point sample measurements of CO and CO_2, mole fraction of sample dried to 32°F saturation.
	21-31	RCO2	R	11	
	32-42	RHC	R	11	Sample measurements of hydrocarbon (as CH_4) and total oxides of nitrogen, mole fraction in net sample.
	43-53	RNOX	R	11	
	54-64	PT	R	11	Probe impact pressure measurement, psia.
	65-75	PS	R	11	Local static pressure, psia.
	76-86	BLOC	R	11	Local effective by-pass ratio.
	87-97	EICOC	R	11	CO emission index of main combustor, lb CO/lb fuel.
INMOLF (LSCK)					
	1-12	XIN	R	1	Initial mole fractions at the beginning of a kinetics step, mols i/mole mix.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
INNAME (JETMIX)					Column vector names - JETMIX input (not used)
	1	NB	I	1	Number of B-input columns.
	2-6	TAB	H	5	Names of input columns.
	7	ND	I	1	Number of D-input columns.
	8-11	TAD	H	4	Names of D-input columns.
INP1 (JETMIX)					Entry control block (JTSTEP).
	1	ENTRY1	L	1	Set T for first entry into JTSTEP.
INPJET (JETMIX)					Jet input parameters
	1	DIAJ	R	1	Primary jet diameter, inches or width.
	2	MJET	R	1	Primary jet reference Mach number.
	3	TJET	R	1	Primary jet reference static temperature, °R.
	4	PTJET	R	1	Primary jet reference stagnation pressure, psia.
	5	VJET	R	1	Primary jet reference velocity, fps.
	6	TIJET	R	1	Primary jet turbulence intensity (relative to VJET).
	7	PE	R	1	Boundary condition - static pressure, psia.
	8	VE	R	1	Boundary condition - external velocity, fps.
	9	ME	R	1	Boundary condition - external Mach number.
	10	TIE	R	1	Boundary condition - turbulence intensity
	11	TE	R	1	Boundary condition - static temperature, °R.
	12	AXI	L	1	T = axisymmetric jet. F = plane or 2-D jet.
	13	NJ	I	1	Index of nozzle lip mesh point.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
INPJET (cont'd.)					
	14	NMAX	I	1	Number of mesh points allowed before grid is refined.
	15-114	X	R	100	Table of primary X/D values.
	115-214	XPRN	L	100	Reference - X Table of print stations. T = Print corresponding X station.
	215	GAM	R	1	Not used.
	216	RG	R	1	Gas constant ft lbf/sec^2 °R.
	217	PR	R	1	Prandtl number C_p μ/K.
	218	PRT	R	1	Turbulent Prandtl number, C_p μ_t/K_t.
	219	SC	R	1	Sutherland constant (viscosity formulation).
	220	TREF	R	1	Reference temperature for viscosity calculation.
	221	MUREF	R	1	Reference viscosity at TREF.
IOFILE (MAIN)					I/O file indication.
	1	TAPIN	L	1	T = input file. T = output file.
JET (JETMIX)					Properties of jet along axis of symmetry or on centerbody.
	1-100	B	R	100	Non-dimensional width of mixing zone.
	101-200	UC	R	100	Velocity (u/u_J).
	201-300	TC	R	100	Static temperature (T/T_J).
	301-400	TIC	R	100	Turbulence intensity (u').
	401-500	PTC	R	100	Stagnation pressure ratio $\frac{(P_T - P_E)}{(P_{Tjet} - P_E)}$
	501-600	WJ	R	100	Relative entrained flow W/W_J.
	601-700	YJ	R	100	Y location of jet edge.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
JET1 (JETMIX)					Jet parameters.
	1	EJET	R	1	Reference turbulent kinetic energy for primary jet, Btu/lbm.
	2	FLOWJ	R	1	Normalizing jet flow rate at initial station.
	3	NXN	I	1	Current calculation station index.
	4	TTO	R	1	Jet stagnation temperature, °R.
JET2 (JETMIX)					Station stagnation temperature.
	1-100	TTC	R	100	Stagnation temperature ratio on jet axis or center body. $\frac{T_{T℄} - T_e}{T_{To} - T_e}$
JET3 (JETMIX)					Control common for added stations. Confined mixing.
	1	NV	I	1	Added station counter.
	2	STADD	L	1	T = skip property calculation at added station.
	3	STATF	L	1	T = do not add to "additional station counter" (NV)
JETDAT (PREJET, EMIS)					Mean properties of flow at nozzle exit survey points. Also info transmitted from MXFLUT to HYCARB.
	1	NPTS	I	1	Number of survey points (eleven max.; point NPTS+1 is ambient air).
	2-13	RAD	R	12	Radial location, ft.
	14-25	TS	R	12	Static temperature, °R.
	26-37	U	R	12	Velocity, fps.
	38-49	SPV	R	12	Specific volume, ft^3/lbm.
	50-61	MWT	R	12	Mean molecular weight, lbm/mole.
	62-73	CP	R	12	Specific heat, Btu/lbm °R.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
JETDAT (cont'd.)					
	74-85	FUEL	R	12	Fuel concentration, lb fuel/ lb mix
	86-97	SPALDG	R	12	Heterogeneity parameter, (lb fuel/lb mix)^2.
	98-109	TKE		12	Not used.
	110-121	TFR	R	12	Mass of fuel transported into the hot part of a tube from the hot parts of all tubes in a mixing step.
	122	TFL	R	1	Mass of fuel transported into the hot part of a tube from the cold part of that tube in a homogenization step.
	123-134	TIM1	R	12	Cumulative residence time of all tubes at previous computation station.
	135	TIM2	R	1	Cumulative residence time of a tube at present computation station.
	136-145	OTHER		10	Unused space.
JETTWO (JETMIX)					Coannular jet input parameters.
	1	TWO	L	1	T = Coannular jet. F = Single jet.
	2	DIAO	R	1	Secondary jet diameter, inches or width.
	3	MJETO	R	1	Secondary jet reference Mach number.
	4	TJETO	R	1	Secondary jet reference static temperature, °R.
	5	VJETO	R	1	Secondary jet reference velocity, fps.
	6	PTJETO	R	1	Secondary jet reference stagnation pressure, psia.
	7	TIJETO	R	1	Secondary jet turbulence intensity (relative to VJET).
	8	NJO	I	1	Index of secondary nozzle lip mesh point.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
JMIXIC (LMXF)	2199				Variables stored on various output files but not used in the program reading the file. Used also as dummy or temporary storage during input/output operations.
KEYS (GMOD, JETMIX, LMXF, PREJET, SPALDG)					File record indices.
	1-10	KEYA	I	10	File record indices for identification.
	11-20	KEYB	I	10	Not used.
	21-30	RODA	I	10	Not used.
	31-40	RODB	I	10	Not used.
KININS (LSCK)					Data storage for KINET.
	1	XMWUN	R	1	Unreacted molecular weight of fuel-air mixture, lbm/lb mole.
	2	XMWC	R	1	Fully recombined molecular weight of combustion products, lbm/lb mole.
	3	TCONST	L	1	Not used. Set F.
	4	CONER	L	1	Not used. Set T.
	5	XNOI	R	1	Initial mole.
KOUT (EMIS)					Information used by GCKP (see Reference 3).
	1-20	TITLE	H	20	Page heading text, same as in OPCTRL
	21	UNITI	H	1	Input units (= FPS)
	22	UNITO	H	1	Output units (= FPS)
	23	CONC	L	1	Output composition will be converted to mole fractions (= .TRUE.).
	24	EXCHR	L	1	Energy exchange rates rather than conversion rates will be output (= .FALSE.).
	25-54	DELH	R	30	Heat of reaction.
	55	FPS	H	1	"FPS"
	56	SI	H	1	"SI"
	57	DBUGO	L	1	Print intermediate output (= .FALSE., change via Namelist PROB).

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
LLTERP (JETMIX)					Transition region scale selection.
	1	LTERP	L	1	T = linear variation of turbulence scale. F = Exponential variation of turbulence scale.
LTUS (LGCK)					File codes used in NASA GCKP program (Reference 3). Not used in PLUMOD.
	1	LTHM	I	1	File containing polynomial coefficients for species thermo properties.
	2	LDAT	I	1	Scratch file for storing input data.
MERGET (JETMIX)					Coannular jet - Merge information.
	1	MER	L	1	Merge indicator - Set to T on merge. Set to F after processing merge station.
	2	MERSTP	L	1	Merge indicator - Set to T on merge.
	3	XMRG	R	1	X location of jet merge
MISC (JETMIX)					Diagnostic print block.
	1-20	PM	R	20	Diagnostic print indicators.
MIXER (JETMIX)					Confined jet - input parameters.
	1	MIX	L	1	T = confined mixing. F = free mixing
	2-101	RDD	R	100	Outer duct radius or Y location, inches.
	102-201	XD	R	100	X station location, inches.
	202	CF	R	1	Not used.
	203	YR	R	1	RD nondimensionalized by DIAJ/2.
MIXPRP (JETMIX)					Confined mixing - inviscid stream parameters (not used).
	1-100	MA2	R	100	Mach number.
	101-200	VE2	R	100	Velocity, fps.

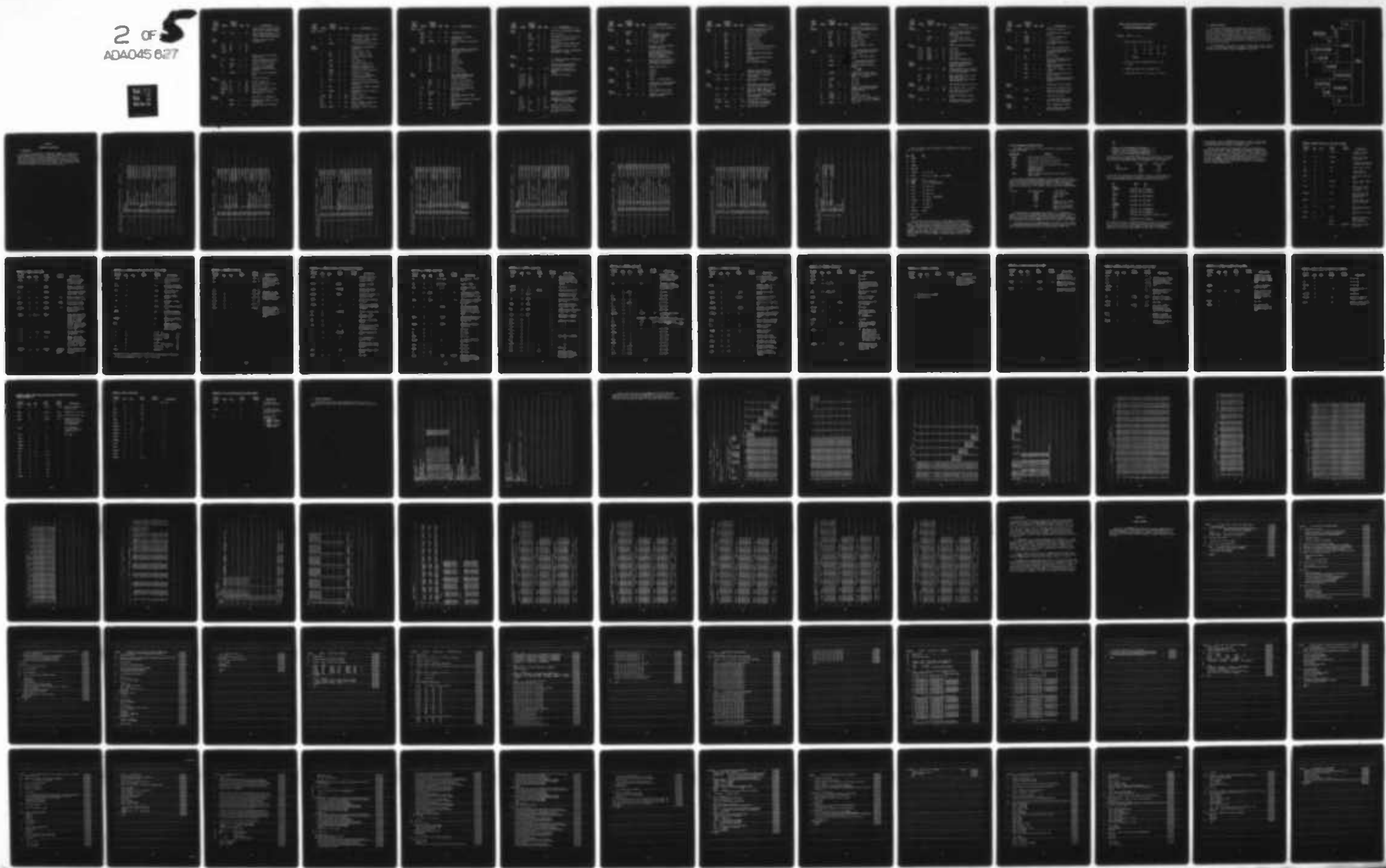
Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
MIXPRP (cont'd.)					
	201-300	TE2	R	100	Static temperature, °R.
	301-400	TWC	R	100	Wall temperature, °R.
MOLES (JETMIX)					Species mole fraction storage (working).
	1-1200	ALJ	R	100, 12	Species mole fractions i = 1, 2.
MOLUP (CPLINK)					Species mole fraction storage (working) TKE working storage.
	1-1200	ALXU	R	100, 12	Species mole fractions at upstream station.
	1201-2400	DALXU	R	100, 12	Normal deviative of species mole fractions.
	2401-2600	DTKE	R	200	Normal deviative of turbulent kinetic energy.
NOXTRA (LSCK)					Subroutine NOX2B
	1	DNOXDT	R	1	Rate of NOX production/consumption $\frac{dNO_x}{dt}$.
OPCTRL (PREJET, EMIS)					Output control information.
	1-20	TITLE	H	20	Six-character Hollerith words transmitted to page headings.
	21-50	PRINT	R	30	Axial stations (feet) at which emissions summation and printout are desired.
OPTS (LGCK)					GCKP input options (Reference 3). See NAMBLK.
	1	VERSI	H	1	"TIME", the independent variable.
	2	TIMEV	H	1	"TIME"
	3	VERSA	H	1	"AREA", the assigned variable.
	4	AREAV	H	1	"AREA"
	5	ELIM	L	1	Automatically eliminate from error consideration species with nonrepresentative errors.
	6	TCON	L	1	Hold temperature constant.
	7	RHOCON	L	1	Hold density constant.
	8	IPRCOD	I	1	Input type indicator (not used).

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
OUTMIX (JETMIX)					Output common block.
	1	NXORIG	I	1	Number of calculation stations.
PARAM (JETMIX)					Communication common - Coefficient of different equations (DFEQ).
	1-200	AL	R	200	Coefficient α.
	201-400	BE	R	200	Coefficient β.
	401-600	GM	R	200	Coefficient γ.
	601-800	EPS	R	200	Coefficient e.
	801-1000	DL	R	200	Coefficient δ.
	1001-1200	VAR	R	200	Dependent variable $\left\{\begin{matrix} U \\ \theta \\ E \\ K_i \end{matrix}\right\}$.
	1201-1400	SM1	R	200	Independent variable (ψ) at upstream station.
	1401	NM1	I	1	Number of mesh points at upstream station.
	1402-1601	SM	R	200	Independent variable (ψ) at downstream (current) station.
	1602	NM	I	1	Number of mesh points at downstream station.
	1603	DX	R	1	Streamwise step size (ΔX).
	1604	B1	R	1	Boundary condition coefficients applied at mesh point 1.
	1605	C11	R	1	
	1606	D1	R	1	
	1607	AN	R	1	Boundary condition coefficients applied at mesh point NM.
	1608	BN	R	1	
	1609	CN	R	1	
PARAM1 (CPLINK)					Communication common - coefficients of difference equations (DFEQ)
	1-200	ETA	R	200	Coefficient η.

AD-A045 627 GENERAL ELECTRIC CO CINCINNATI OHIO AIRCRAFT ENGINE GROUP F/G 21/5
DEVELOPMENT OF EMISSIONS MEASUREMENT TECHNIQUES FOR AFTERBURNIN--ETC(U)
OCT 75 W C COLLEY, D R FERGUSON, M A SMITH F33615-73-C-2047

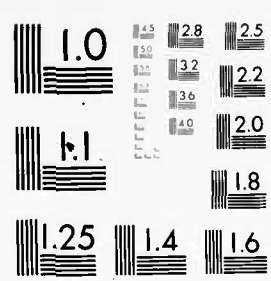
UNCLASSIFIED R75AEG459 AFAPL-TR-75-52-SUPPL-2 NL

2 OF 5
ADA045 627



2 OF

ADA045627



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
PORIDE (EMIS)					Input storage to override print stations set in PRELIM overlay.
	1-30	PSTA	R	30	Print station over-ride. Values will be used in deference to those in OPCTRL (jet diameters).
PQRE (LGCK)					Internal storage for GCKP (reference 3).
	1-28	PK	?	28	
	29-56	QK	?	28	
	57-84	RK	?	28	
	85-112	E	?	28	
PRIN (LGCK)					Output control for NASA GCKP (reference 3).
	1-50	PRINT	R	50	Values of independent variable at which output is desired.
	51	NPRNTS	I	1	Number of values of PRINT.
	52	END	R	1	Last value of independent variable at which output is desired.
	53	EVSTEP	L	1	Print after every integration step.
PROF (JETMIX)					Profiles across jet.
	1-200	PSI	R	200	Stream function (ψ).
	201-400	Y	R	200	Normal coordinate Y = 2Y/DJ.
	401-600	UD	R	200	Velocity ($U = u/u_J$).
	601-800	THD	R	200	Static temperature ($\theta = T/T_J$).
	801-1000	ED	R	200	Turbulent kinetic energy ($E = e/e_J$).
PROPJ2 (JETMIX)					Coannular jet - properties.
	1	MACHO	R	1	Reference Mach number for scale calculation.
	2	REFLO	R	1	Reference scale for secondary jet.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
PROPJ2 (cont'd.)					
	3	YI	R	1	Y coordinate - edge of mixing zone of primary jet.
	4	YO	R	1	Y coordinate - edge of mixing zone of secondary jet.
	5	MERGE	L	1	T = downstream of merge point.
PROPJT (JETMIX)					Transport and thermodynamic properties.
	1	P	R	1	Static pressure, psia.
	2	PRL	R	1	Prandtl number, $C_p\ \mu/K$.
	3	PRT	R	1	Turbulent Prandtl number, $C_p\ \mu_t/K_t$.
	4	RGAS	R	1	Gas constant RG.
	5	SC	R	1	Sutherland constant, °R.
	6	TREF	R	1	Reference temperature for viscosity calculation, °R.
	7	MUREF	R	1	Viscosity at TREF, lbm/ft sec.
	8	MACH	R	1	Reference Mach number.
	9	XLC	R	1	X location of disappearance of jet potential core.
	10	REFL	R	1	Reference scale of turbulence.
	11	C	R	1	Constant for turbulent viscosity calculation.
	12	CHI	R	1	Constant for turbulent viscosity calculation.
	13	RNORM	R	1	Normalizing Reynold's number of turbulence
	14-213	RHO	R	200	Density, lbm/ft^3.
	214-413	MUL	R	200	Laminar dynamic viscosity (μ) lbm/ft sec.
	414-613	KCP	R	200	Ratio of thermal conductivity to heat capacity, K/C_p.
	614-813	MUEFF	R	200	Turbulent + laminar viscosity ($\mu + \mu_t$).

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
PROPJT (cont'd.)					
	814-1013	XLN	R	200	Turbulence scale.
	1014-1213	DK	R	200	Dissipation (turbulence) coefficient.
	1214-1413	RETURB	R	200	Reynold's number of turbulence $\frac{\rho v_e L_t}{\mu}$.
PROPR (LSCK)					Subroutine KINET - thermodynamic quantities (equilibrium) (not used).
	1	PR	R	1	Pressure.
	2	HHR	R	1	Enthalpy.
	3	TR	R	1	Temperature.
	4	FCR	R	1	Fuel/air ratio.
	5	RHOR	R	1	Density.
	6	RR	R	1	Gas constant.
	7	WMTR	R	1	Molecular weight.
PSEQ (LSCK)					SCKP - data storage. Communication common between kinetics routines.
	1	FOA	R	1	Fuel/air ratio, lbm/lbm.
	2	BETS	R	1	Combustion efficienty, (β).
	3	TP	R	1	Reaction temperature, °R.
	4-19	X	R	16	Constituent mole fractions, mols i/mole mix.
	20	DHQDMW	R	1	Not used.
	21	TEQ	R	1	Equilibrium reaction temperature, °R.
	22	BEQ	R	1	Equilibrium combustion efficiency, (β_{eq}).
	23	XMWT	R	1	Mixture molecular weight, lbm/lb mole.
	24	HNEQ	R	1	Not used.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
PSEQX (LSCK)					Fuel and air communication commo
	1	HQC	R	1	Fuel H/C ratio.
	2	HUM	R	1	Air humidity, lbm H_2O/lbm dry ai
	3	CO2AIR	R	1	Air - CO_2 concentration, mol CO_2/mol air.
	4	MAIR	R	1	Air molecular weight, lbm/lb mole
	5	FS	R	1	Stoichiometric fuel/air ratio, lbm/lbm.
	6	FUELMW	R	1	Fuel molecular weight, lbm/lb mol 1gm atom of C - basis.
RATIO (JETMIX)					
	1	AMBTO	L	1	T = ambient stagnation temperatur set. (JETMIX)
REAC (LGCK)					Chemical reaction information used by GCKP (Reference 3).
	1-120	LSR	I	4,30	Code number of species involved in each reaction (see BLKRDA).
	121-150	XX	?	30	Internal GCKP variable.
	151-180	RATE	?	30	"
	181-210	LKEQ	?	30	"
	211-240	DLKEQ	?	30	"
	241-270	MM	?	30	"
	271	LR	?	1	"
RRAT (LGCK)					Reaction rate coefficients and third-body efficiencies for GCKP (Reference 2).
	1-30	A	R	30	Coefficients for Arrhenius expressions for reaction rates Rate = $AT^N \exp(-E/RT)$.
	31-60	N	R	30	
	61-90	EACT	R	30	
	91-120	B	?	30	Internal GCKP variable.
	121-870	M	R	25,30	Third-body efficiencies.
	871	ALLM1	L	1	All third-body efficiencies are 1.0.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
RSTART (JETMIX)					Calculation restart parameters.
	1	MIXPRE		1	Not used.
	2	NREG	I	1	1 = potential core region. 2 = transition region. 3 = similar region.
	3	NRES	I	1	Restart (X) station.
	4	RESTRT	L	1	T = restart case.
SCALED (JETMIX)					Turbulence scale from species diffusion equation.
	1	SCLD	L	1	T = use mole fraction of ambient species for determination of mixing zone width reference dimension. F = use velocity profile.
	2	ALXLIM	R	1	Limiting value for detection of inner edge of mixing zone in the "potential core region".
SCALER (JETMIX)					JETMIX input scalers.
	1	SP	R	1	Pressure.
	2	SV	R	1	Velocity
	3	SLEN	R	1	Length.
SETNEW (JETMIX)					JETMIX Jet edge common.
	1	LKK	I	1	Mesh index - ≤3 forces addition of mesh point at jet edge.
	2	LCOR	I	1	Not used.
SING (JETMIX)					Dummy JETMIX input common.
	1-43	SSD	R	43	Temporary input storage for single cell input.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
SINT (LGCK)					Integration step size controls for GCKP (Reference 3).
	1	HMIN	R	1	Minimum step size, sec.
	2	HINT	R	1	Initial step size.
	3	HN	R	1	Last step size.
	4	HNP1	R	1	Next step size.
	5	HMAX	R	1	Maximum step size.
	6	NH	I	1	Number of steps since last print.
	7	AVH	R	1	Average step size since last print.
	8	EMAX	R	1	Maximum tolerable integration error.
	9	ERRN	R	1	Estimated error from proposed next step size.
	10	JCV	?	1	GCKP internal variable.
	11	KOUNT	?	1	"
	12	ERRP	?	.	"
SNMW (EMIS)					Chemical species names and molecular weights (see BLCK).
	1-150	ALSP	H	2,75	Specie names, 12 characters, left-justified.
	151-225	ALMW	R	75	Specie molecular weights ($M_o \equiv 16$).
SPEC (LGCK)					Miscellaneous checmical reaction info used by GCKP (Reference 2).
	1-60	SNAM	H	2,30	Names of chemical species copied from block SNMW starting at SNAM (1,4). See also NAMBLK.
	61-85	MW	R	25	Specie molecular weights, copied from block SNMW.
	86-110	W	?	25	Internal GCKP variable.
	111-860	STOIC	R	25,30	Stoichiometric coefficients for reactions (see BLKST).
	861-1610	OMEGA	?	25,30	Internal GCKP variable.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
STA2 (JETMIX)					Confined jet - iterated conditions at downstream station.
	1	MACH2	R	1	Mach number.
	2	TS2	R	1	Static temperature, °R.
		SS2	R		Sonic velocity, fps.
	3	V2	R	1	Velocity, fps.
	4	RHO2	R	1	Density, lbm/ft^3.
	5	DPDX2	R	1	Pressure gradient, psi/in (dP/dx).
STCTRL (EMIS)					Control information for mixing/homogenization and kinetics computations.
	1	LSTA	I	1	Current calculation station index.
	2	FINAL	L		T = calculation has reached a streamwise print station.
	3	CHEMK	I	1	Chemical kinetics selection switch 1 = GCKP 2 = SCKP
	4	FIRSTM	L	1	Indicator for initial mixing step. Set T on first entry to MXFLUT. F thereafter.
	5	FIRSTC			Not used.
	6	SC			Not used.
	7	DXC	R	1	Not used.
	8	NIST	I	1	Number of intermediate mixing/homogenization steps between print locations.
	9	POUT1	R	1	Not used.
	10	ALFLIM	R	1	Limiting mixing rate.
	11	CMIXST	R	1	Mixing step size control parameter. $\alpha_{min} > \alpha_{lim}$ ΔX = CMIXST*
	12-17	DUMST	R	6	Not used.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
SUPER (JETMIX)					Supersonic jet - Sonic line bloc
	1	SUPC	L	1	T = no supersonic points Reset to F after station processing.
	2	SUPSTP	L	1	T = no supersonic points.
	3	XSUP	L	1	X station where flow is fully subsonic.
TAG (JETMIX)					JETMIX Identification block
	1-10	NAME	I	10	User name.
	11-20	TITLE	R	10	Case title.
	21-30	IDENT	I	10	Case identification.
	31-40	ADDRES	R	10	User address.
	41-50	IDENT1	I	10	Extra case identification.
TCOF (PREJET, EMIS)					Polynomial coefficients for species thermo properties (NASA TN D-4097).
	1-350	C,TC	R	7,2, 25	Seven coefficients for high and low temperature ranges for each of up to 25 species.
	351	TLOW	R	1	High temperature range is from TMID to THI (°K).
	352	TMID	R	1	Low temperature range is from
	353	THI	R	L	TLOW to TMID (°K).
THERM (JETMIX)					Thermodynamic properties (vs Y).
	1-200	GMC	R	200	Heat capacity ratio $\gamma = C_p/C_v$.
	201-400	CP	R	200	Heat capacity at constant pressure, Btu/lbm °R.
THRST (JETMIX)					Integrated momentum (thrust) block.
	1-100	WV	R	100	Integrated momentum (thrust) at each station, ft lbm/sec^2.

Block Name (Link)	Words	Typical Variable Name	Type	Dim.	Description
TROUBL (MAIN)					PLUMOD error switches.
	1	ERR	L	1	T = program error (normally fatal).
	2	ERRMAJ	L	1	T = fatal program error.
	3	INERR	L	1	T = input error.
	4	PRERR	L	1	T = previous program error (not used).
UMESH (JETMIX)					Mesh or grid common block.
	1	MCHANG	L	1	T = do not redistribute mesh when the potential core disappears.
	2	CK	R	1	Mesh constant.
	3	DY1	R	1	ΔY (1st - 2nd mesh point) at point where potential core disappears.
	4	NMSH	I	1	Number of mesh points in the redistributed grid.
	5	CXPC	R	1	ΔX step size (potential core region).
	6	CXTP	R	1	ΔX step size (transition and similar region).
	7	NRED	I	1	Number of mesh points discarded when maximum points (NM) are reached.
XISAVE (EMIS)					Indexing info used by READT2.
	1	I	I	1	READT profile storage index.
	2	ISTAR	I	1	Number of last set of profile data read from JETMIX file.
XPRIN (JETMIX)					Profile print switch.
	1	DPRIN	L	1	T = print JETMIX profiles at every step (not recommended).
YOFXI (JETMIX, LSCK)					YOFX interpolation common
	1	IE	I	1	Saves interpolation index for use on the next entry.

Table 2. Key to Subscripts for Variables in Common Blocks/GASCMP/and/GASTMW/.

Example: CONC (I, J, K, L)

I Species identification in order:

1	H	O	H_2	O_2	OH
6	H_2O	CO	CO_2	N_2	A
11	NO	N	HO_2	NO_2	NH_3
16	$C_{10}H_{10n}$				

J Portion of 2-part heterogeneous gas (1 = hot, 2 = cold)

K NEWNET tube number (Last is ambient air)

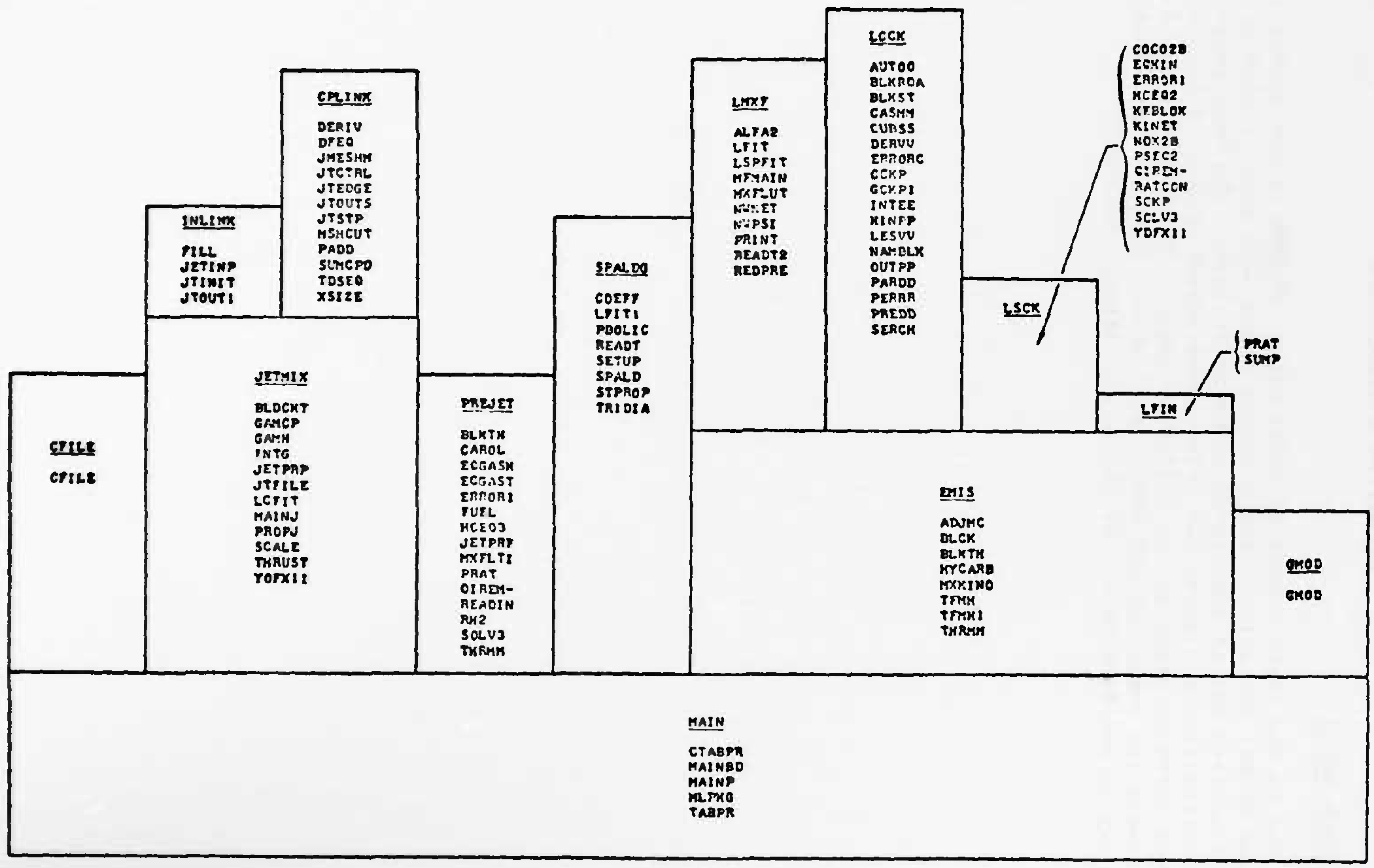
L Computation station (1 = previous, 2 = present)

2.3 Overlay Structure

To conserve computer memory, the program is run as an overlay job. The link structure is depicted in Table 3. Link names are underlined. The content of each link is described by deck labels, corresponding to the sub-program names in Sections 2.1 and 4.0. System library subroutines are not shown. The vertical heights of the bo es in Table 3 represent words of memory, roughly to scale. The horizontal axis corresponds to different times during execution of the program. The links are shown in the order that they exist on the loader input file (Section 3.1), not necessarily in the order in which they are executed.

The subprograms in each link are listed in alphabetical order in Table 3; however, it is important that the first decks in links JETMIX, PREJET, SPALDG, and EMIS be MAINJ, READIN, SPALD, and MISKINO, respectively.

Table 3. PLUMOD Overlay Diagram.



SECTION 3

OPERATING INSTRUCTIONS

3.1 Deck Setup

Due to the large number of subprograms, PLUMOD is not loaded from program decks, but rather from a loader input (R*) file created by the Object Library Editor and stored on magnetic tape. The file contains job control language cards and relocatable object decks. A list of the control cards is given on the following pages. The $ OBJECT and $ DKEND cards mark the beginning and end of each object deck. If necessary, the file may be modified by use of the Object Library Editor.

```
A6000 01  06-30-75   12.709   OBJECT LIBRARY EDITOR ACTIVITY REPORT          VERSION 740320              PAGE      1
CONTENTS OF NEW OBJECT LIBRARY

              $     LOWLOAD 16
              $     OPTION  FORTRAN
              $     LIBRARY SL
              $     OBJECT  MAIN PROGRAM--EMIS DRIVER                              F08.566063075MAINP000
                            ------
              $     DKEND                                                          08.566063075MAINP025
              $     OBJECT     MAIN BLOCK DATA                                     F17.367061775MAINB000
                               TROUBL
              $     DKEND                                                          17.367061775MAINB003
              $     OBJECT     TABLE PRINTOUT                                      F18.211062575TABPRT00
                               TABPRT
              $     DKEND                                                          18.211062575TABPRT16
              $     OBJECT     BLOCK DATA FOR TABPRT                               F18.214062575CTABPR00
                               CTABPR
              $     DKEND                                                          18.214062575CTABPR03
              $     USE      MOVE
              $     LINK     CFILE
              $     OBJECT    MERGE OR SPLIT FILES                                 F12.698063075CFILE000
                              MAIN
              $     DKEND                                                          12.698063075CFILE000
              $     LINK     JETMIX,CFILE
              $     OBJECT    MAIN JETMIX SUBROUTINE                               F10.897062575MAINJ000
                              MAIN
              $     DKEND                                                          10.897062575MAINJ008
              $     OBJECT      PROPERTIES OF JET                                  F19.324103174JETPRP00
                               JETPRP
              $     DKEND                                                                   JETPRP46
              $     OBJECT     STORAGE OF AERO-DATA FILE FOR PLOT OR N             F10.902062575JIFILE00
                               JIFILE
              $     DKEND                                                          10.902062575JIFILE05
              $     OBJECT     CALCULATE NET THRUST--LBF                           F20.419072070THRUST00
                               THRUST
              $     DKEND                                                                   THRUST39
              $     OBJECT    FUNCTION  GAMM  (T.--P.H. HECK                       F08.011052970GAMM0000
                              GAMM
              $     DKEND                                                                   GAMM0005
              $     OBJECT     COMPUTE GAMMA, CP----   (P.H. HECK)                 F08.013052970GAMCP000
                               GAMCP
              $     DKEND                                                                   GAMCP007
              $     OBJECT     BLOCK DATA FOR TURBULENT PARAMETERS                 ,05.260101670BLOCKT00
                               CPROP
              $     DKEND                                                                   BLOCKT00
              $     OBJECT  PROPERTIES OF A LAMINAR OR TURBULENT JET               F13.979041875PROPJ000
                               PROPJ
              $     DKEND                                                          13.979041875PROPJ034
              $     OBJECT    TURBULENCE SCALE/SINGLE,COANNULAR/COPLAN             F10.684100174SCALE000
                              SCALE
              $     DKEND                                                               TS    SCALE015
              $     OBJECT     INTEGRAL OF Y*DX--TRAPEZOIDAL/UNEQUAL X.            F15.780070769INTG0000
                               INTG
```

```
A6000 01  06-30-75   12.709   OBJECT LIBRARY EDITOR ACTIVITY REPORT          VERSION 740320              PAGE     2
CONTENTS OF NEW OBJECT LIBRARY

          $    DKEND                                                           INTG0006
          $    OBJECT                                                     072468LCFIT000
                         LCFIT
          $    DKEND                                                           LCFIT028
          $    OBJECT                                                .16.596103169YOFX1100
                         YOF
          $    DKEND                                                           YOFX1110
          $    OBJECT    BLOCK DATA FOR TABPRT                       F16.275040170CTABPR00
                         CTABPR
          $    DKEND                                                           CTABPR03
          $    LINK    INLINK
          $    OBJECT    INITIALIZE NEW PROBLEM                      F13.989041875JTINIT00
                         JTINIT
          $    DKEND                                                  13.989041875JTINIT23
          $    OBJECT    JET--INPUT ROUTINE                          F08.718062675JETINP00
                         JETINP
          $    DKEND                                                  08.718062675JETINP34
          $    OBJECT     JET/OUTPUT AT STATION 1                    F17.038021174JTOUT100
                         JTOUT1
          $    DKEND                                                           JTOUT141
          $    OBJECT                                                F15.927052069FILL0000
                         FILL
          $    DKEND                                                           FILL0011
          $    OBJECT                                                   080168ISORT000
                         ISORT
          $    DKEND                                                           ISORT009
          $    OBJECT                                                G17.202020470FMNL1000
                         FMNL1
          $    DKEND                                                           FMNL1005
          $    LINK    CPLINK,INLINK
          $    OBJECT     MAIN JET CONTROL ROUTINE                   F17.947062575JTCTRL00
                         JTCTRL
          $    DKEND                                                  17.947062575JTCTRL20
          $    OBJECT    JET--SOLUTION ROUTINE                       F07.859100574JTSTP000
                         JTSTEP
          $    DKEND                                                           JTSTP097
          $    OBJECT   DFEQ COMPATIBLE DERIVATIVE                   .11.668102970DERIV000
                         DERIV
          $    DKEND                                                           DERIV000
          $    OBJECT     LOCATE EDGE OF JET       NEW Y CO-ORDI     F08.211052075JTEDGE00
                         JTEDGE
          $    DKEND                                                  08.211052075JTEDGE59
          $    OBJECT     SUMMARY PRINT OF JET PROPERTIES            F09.224101273JTOUT300
                         JTOUT3
          $    DKEND                                                           JTOUT361
          $    OBJECT    DIFFUSION FLUX TERM FOR ENERGY EQUATION     F04.815092673SUMCPD00
                         SUMCPD
          $    DKEND                                                           SUMCPD06
          $    OBJECT     MESH REFINEMENT ROUTINE                    F16.736060870MSHCUT00
                         MSHCUT
```

CONTENTS OF NEW OBJECT LIBRARY

```
$   DKEND                                                          MSHCUT30
$   OBJECT     X-STEP SIZE CONTROL                     .10,646090869XSIZE000
               XSIZE
$   DKEND                                                          XSIZE010
$   OBJECT     ADDITION OF MESH POINTS AT JET EDGE     F19,040020874PADD0000
               PADD
$   DKEND                                                          PADD0009
$   OBJECT     MESH REDISTRIBUTION FOR TRANS. REGION   F06,380071769JMESHM00
               JMESHM
$   DKEND                                                          JMESHM08
$   OBJECT     GEN. ROUTINE-DIFFUSION EQUATION EQN--CENT F11,690021178DFEQ0000
               DFEQ
$   DKEND                                                          DFEQ0008
$   OBJECT                                                  031369TDSEQ000
               TDSEQ
$   DKEND                                                          TDSEQ011
$   EQUATE   ERROR1/AITER1.AITER2/
$   LINK     PREJET.JETMIX
$   OBJECT     INPUT ROUTINE FOR EMISSIONS PROG.       F18,223082874READIN00
               READIN
$   DKEND                                                          READIN00
$   OBJECT     GENERATES INITIAL PROFILES FOR -JETMIX- F10,912062575JETPRF00
               JETPRF
$   DKEND                                             10,912062575JETPRF62
$   OBJECT     INITIATE HETEROGENEOUS GAS MODEL AT EACH P F10,811042275MXFLTI00
               MXFLTO
$   DKEND                                             10,811042275MXFLTI71
$   OBJECT     IMPACT/STATIC PRESSURE RATIO   PERFECT GAS F10,823100373PRAT0000
               PRAT
$   DKEND                                                          PRAT0005
$   OBJECT     HYDROGEN CONCENTRATION ESTIMATED FROM CO M F10,864100373RH200000
               RH2
$   DKEND                                                          RH200006
$   OBJECT     QIRFM QUITS                             F10,819100373ERROR100
               ERROR1
$   DKEND                                                          ERROR109
$   OBJECT     FUEL-AIR RATIO AND EMISSION INDICES FROM C F10,877100373CAROL000
               CAROL
$   DKEND                                                          CAROL014
$   OBJECT     THERMO PROPERTIES OF RAW FUEL VAPOR     F10,994100373FUEL0000
               FUEL
$   DKEND                                                          FUEL0013
$   OBJECT     THERMO PRPTYS OF EQLBRM COMBN GAS - T INDE F18,456082974EQGAST00
               EQGAST
$   DKEND                                                          EQGAST16
$   OBJECT     THERMO PRPTYS OF EQLBRM COMBN GAS - H INDE F18,233082874EQGASH00
               EQGASH
$   DKEND                                                          EQGASH00
$   OBJECT     FULL-EQUILIBRIUM COMPOSITION CH(N)-AIR-H. F18,078082175HCEQ3000
               HCEQ2
```

CONTENTS OF NEW OBJECT LIBRARY

```
$     DKEND                                                        15,074042175MCEQ3051
$     OBJECT   SOLVE 3 LINEAR EQUATIONS  A(IJ)*X(J)=B(I)           F17,804031474SOLV3000
                  SOLV3
$     DKEND                                                                 SOLV3012
$     OBJECT   CALCULATES (DIMENSIONLESS) THERMODYNAMIC P          F12,985120171THRMM000
                  THRM
$     DKEND                                                                 THRMM020
$     OBJECT   -BLKTH-      BLOCK DATA  ( THERMO )                 F18,318072574BLKTH000
                  TCOF
$     DKEND                                                                 BLKTH015
$     OBJECT    QUADRATIC INTERPOLATION ROOT EVALUATION            F11,611070572QIREM-00
                  QIREM
$     DKEND                                                                 QIREM-28
$     LINK     SPALDG,PREJET
$     OBJECT   -SPALD-  MAIN SUBROUTINE FOR CALC. OF -G-           F10,918062575SPALD000
                  SPALD
$     DKEND                                                        10,918062575SPALD086
$     OBJECT    READS -JETMIX- OUTPUT TAPE                         F08,575063075READT000
                  READT
$     DKEND                                                        08,575063075READT087
$     OBJECT    CALCULATES FLUID PROPERTIES AT EACH POINT          F21,050103073STPROP00
                  STPROP
$     DKEND                                                                 STPROP09
$     OBJECT    COMPUTES SOLUTION OF PARABOLIC PDE                 F21,084103073PBOLIC00
                  PBOLIC
$     DKEND                                                                 PBOLIC15
$     OBJECT    SETS-UP FLOW PROPERTIES FOR INTERPOLATION          F18,231062575SETUP000
                  SETUP
$     DKEND                                                        18,231062575SETUP025
$     OBJECT    EVALUATES COEFFICIENTS FOR -G- EQUATION            F21,048103073COEFF000
                  COEFF
$     DKEND                                                                 COEFF009
$     OBJECT      LINEAR FIT INTERPOLATION  .                      F19,208120871LFIT1000
                  LFIT1
$     DKEND                                                                 LFIT1009
$     OBJECT    SOLVES TRI-DIAGONAL SYSTEM OF EQUATIONS            F21,064103073TRIDIA00
                  TRIDIA
$     DKEND                                                                 TRIDIA07
$     LINK     EMIS,SPALDG
$     OBJECT      MAIN SUB.-- MIX, FLUTTER, KINETICS, OUT          F17,937062575MXKIND00
                  MXKIND
$     DKEND                                                        17,937062575MXKIND32
$     USE      CINPUT/4511/
$     USE      CINPJT/115/
$     USE      CSPECI/3/
$     USE      CXLOCA/5/
$     USE      XISAVE/2/
$     USE      CMMNET/212/
$     USE      CAGAIN/2/
$     USE      CREACT/12/
```

CONTENTS OF NEW OBJECT LIBRARY

```
$  USE      CDXSAV/1/
$  USE      CCMPCK/37/
$  USE      CMXFLT/444/
$  OBJECT     IGNITION DELAY FOR RAW FUEL                       F07.970080874HYCAR000
              HYCARB
$  OKEND                                                                     HYCAR014
$  OBJECT   CALCULATE TEMPERATURE FROM ENTHALPY                 F09.546080674TFMH0000
              TFMH
$  OKEND                                                                     TFMH0013
$  OBJECT   CP, T FROM H                                        F09.549080674TFMH1000
              TFMH1
$  OKEND                                                                     TFMH1009
$  OBJECT   CALCULATES (DIMENSIONLESS) THERMODYNAMIC P   F12.965120171THRMM000
              THRM
$  OKEND                                                                     THRMM020
$  OBJECT   *BLKTH*     BLOCK DATA  ( THERMO )              F18.318072574BLKTH000
              1COF
$  OKEND                                                                     BLKTH000
$  OBJECT   *BLCK*      BLOCK DATA  ( SP/MW )                F15.713071174BLCK0000
              SNMW
$  OKEND                                                                     BLCK0014
$  OBJECT     MW OF RAW FUEL// ADJUST C2H4 THERMO COEF   F17.607110273ADJMC000
              ADJMC
$  OKEND                                                                     ADJMC000
$  LINK     LMXF
$  OBJECT   *MFMAIN* MAIN SUB. FOR MXFLUT                    F10.926062575MFMAIN00
              MFMAIN
$  OKEND                                                       10.926062575MFMAIN21
$  OBJECT     READS **JETMIX** TAPE (MXFLUT)                  F12.706063075READT200
              READT
$  OKEND                                                       12.706063075READT200
$  OBJECT   REDISTRIBUTION OF STREAM FUNCTION FROM JET  F21.204062174NMPSI000
              NEMPSI
$  OKEND                                                                     NMPSI011
$  OBJECT   *REDPRE* SUB. TO READ PREJET DATA               F08.597063075REDPRE00
              REDPRE
$  OKEND                                                       08.597063075REDPRE29
$  OBJECT   CONVERT JETMIX PROFILES TO FIXED-NUMBER ST  F09.893092674NMNE7000
              NEMNE7
$  OKEND                                                              NET   NMNE7045
$  OBJECT     DUMMY--CALLS LFIT                                F09.376111473LSPFIT00
              LSPFIT
$  OKEND                                                                     LSPFIT04
$  OBJECT       INTEGRATE OR INTERPOLATE             *L   F11.248051173LFIT0000
              LFIT
$  OKEND                                                                     LFIT0017
$  OBJECT       MXFLUT OUTPUT ROUTINE                       F18.234062575PRINT000
              MPRINT
$  OKEND                                                       18.234062575PRINT012
$  OBJECT                                                     F17.658090574MXFLUT00
```

CONTENTS OF NEW OBJECT LIBRARY

```
                        MXFLUT
        $   OKEND                                                                 MXFLUT75
        $   OBJECT  SOLVE FOR MASS TRANSFER MATRIX                 F09.917081974ALFA2000
                        ALFA2
        $   OKEND                                                                 ALFA2000
        $   LINK    LBCK,LMXF
        $   OBJECT   MAIN DRIVER ROUTINE FOR GCKP-1 CALC.          F19.342062774GCKP0000
                        GCKP
        $   OKEND                                                                 GCKP0012
        $   OBJECT  GENERAL CHEMICAL KINETICS PROGRAM              F13.030062574GCKP1000
                        GCKP1
        $   OKEND                                                                 GCKP1015
        $   OBJECT   STOICHIOMETRIC COEFFICIENTS                   F16.793101973BLKST000
                        SPEC
        $   OKEND                                                                 BLKST023
        $   OBJECT  READ INPUT DATA AND INITIALIZE                 F06.531050374KINPP000
                        KINP
        $   OKEND                                                                 KINPP000
        $   OBJECT  -BLKRDA-     BLOCK DATA    ( REACTION DAT      F15.050110474BLKRDA00
                        LSLRSV
        $   OKEND                                                                 BLKRDA34
        $   OBJECT  -NAMBLK-  BLOCK DATA     ( A/N )               F14.693010874NAMBLK00
                        LIUS
        $   OKEND                                                                 NAMBLK05
        $   OBJECT  GENERAL OUTPUT, UNIT CONVERSIONS, ETC          F10.620032773OUTPP000
                        OUTP
        $   OKEND                                                                 OUTPP211
        $   OBJECT  PERFORM ALL NECESSARY PRE-DERIVATIVE CALCU     F11.569032673PREDD000
                        PRED
        $   OKEND                                                                 PREDD037
        $   OBJECT  COMPUTE ALL DERIVATIVES WRT THE INDEPENDEN     F11.576032673DERVV000
                        DERV
        $   OKEND                                                                 DERVV020
        $   OBJECT  COMPUTE ALL MIXED PARTIAL DERIVATIVES          F20.540032673PARDD000
                        PARD
        $   OKEND                                                                 PARDD073
        $   OBJECT  SET UP AND CALLFOR INTEGRATION                 F13.053110873INTEE000
                        INTE
        $   OKEND                                                                 INTEE025
        $   OBJECT  CHOOSE STEP FORMULA, SET UP AUGMENTED MATR     F12.583103074CASMM000
                        CASM
        $   OKEND                                                         NCREMECASMM019
        $   OBJECT  GEN'L DBL PREC LINEAR EQN SOLVER               F12.957120171LESVV000
                        LESV
        $   OKEND                                                                 LESVV026
        $   OBJECT  RELATIVE ERROR IN INTEGRATION STEP             F19.135110873ERRORC00
                        ERRORC
        $   OKEND                                                                 ERRORC12
        $   OBJECT  PREDICTS ERROR TO BE EXPECTED                  F13.058110073PERRR000
                        PERR
```

CONTENTS OF NEW OBJECT LIBRARY

```
$   DKEND                                                                PERRR012
$   OBJECT  AUTOMATIC ELIMINATION FROM ERROR CONSIDERA   F12,966120171AUTO0000
                AUTO
$   DKEND                                                                AUTO0015
$   OBJECT  COEFFICIENTS FOR CUBIC SPLINE INTERPOLATIO   F12,970120171CUBS5000
                CUBS
$   DKEND                                                                CUBS5035
$   OBJECT  OPTIMAL SEQUENTIAL SEARCH TECHNIQUE          F12,975120171SERCH000
                SEARCH
$   DKEND                                                                SERCH014
$   LINK    LSCK,LGCK
$   OBJECT   MAIN DRIVER FOR SCKP CALCULATION            F19,347062774SCKP0000
                SCKP
$   DKEND                                                                SCKP0022
$   OBJECT    HYDROCARBON KINETICS ROUTINE               F11,791112274KINET000
                KINET
$   DKEND                                                                KINET059
$   OBJECT    EQUILIBRIUM TERMS FOR KINETICS CALCULATI   F11,809112274EQKIN000
                EQKIN
$   DKEND                                                                EQKIN024
$   OBJECT                                               F11,631112274KEBLOK00
                PSEOX
$   DKEND                                                                KEBLOK08
$   OBJECT    FULL-EQUILIBRIUM COMPOSITION CH(N)-AIR-H   F17,595042175HCEQ2000
                HCEQ2
$   DKEND                                                 17,595042175HCEQ2051
$   OBJECT  PSEUDO-EQUILIBRIUM COMPOSITION CH(N)-AIR-H   F11,820112274PSEQ2000
                PSEQ2
$   DKEND                                                         STEM  PSEQ2046
$   OBJECT  SOLVE 3 LINEAR EQUATIONS  A(IJ)*X(J)=B(I)    F17,804031474SOLV3000
                SOLV3
$   DKEND                                                                SOLV3012
$   OBJECT    INTEGRATE NO RATE EQUATIONS                F15,052110474NOX28000
                NOX28
$   DKEND                                                                NOX28014
$   OBJECT      COMPUTATION OF RATE CONSTANTS            F19,353062774RATCON00
                RATCON
$   DKEND                                                                RATCON24
$   OBJECT    FINITE RATE CO/CO2 CHEMISTRY               F08,602063075COCO2800
                COCO28
$   DKEND                                                 08,602063075COCO2812
$   OBJECT   QUADRATIC INTERPOLATION ROOT EVALUATION     F11,611070572QIREM-00
                QIREM
$   DKEND                                                                QIREM-25
$   OBJECT  QIREM QUITS                                  F12,962042375ERROR100
                ERROR1
$   DKEND                                                 12,962042375ERROR109
$   OBJECT                                               F09,091080772YOFX1100
                YOF
$   DKEND                                                                YOFX1109
```

```
CONTENTS OF NEW OBJECT LIBRARY

$  EQUATE   ERROR1/CHEMA/
$  LINK     LFIN,LSCK
$  OBJECT   CALC. AND PRINT RESIDUAL EMISSIONS INDICES  F09.1220116753UMP0000
                SUMUP
$  DKEND                                                09.1220116753UMP0000
$  OBJECT   IMPACT/STATIC PRESSURE RATIO   PERFECT GAS  F10.0231005T3PRAT0000
                PRAT
$  DKEND                                                         PRAT0008
$  LINK     GMOD,EMIS
$  OBJECT   MODIFY INITIAL G PROFILES TO FORCE RAPID    F10.9420625756MOD0000
                GMODEI
$  DKEND                                                10.9420625756MOD0049
$  EXECUTE
$  DISC     07
$  DISC     40
$  FILE     39
$  TAPE     01
$  DISC     02
$  DISC     03
$  DISC     04
EOF ENCOUNTERED ON AR
```

The job control language required to run PLUMOD from the loader input tape is given below:

```
1     8             16
$     SNUMB
$     IDENT
$     USERID
$     EXECUTE
$     LIMITS        90, 41K, -4K
$     TAPE          R*, R * D,, 35726  ,, R * PLUMOD
$     SYSOUT        P*
$     MASS          H *, H *R, 60R
$     PRMFL         SL, R, R, AEG-LIB/SC-LB
$     FILE          01, FIR, 10L
$     FILE          02, F2R, 10L
$     FILE          03, F3R, 10L
$     TAPE          04, T4D,,,, CASE-DESCR
$     TAPDATA       04, $R030
$     FILE          39, F8R, 1L
$     FILE          40, F9R, 1L
$     INCODE        IBMF
  (Input data)
$     ENDJOB
```

The user must supply information on the $ IDENT and $ USERID cards. The R * tape number was current on date of publication, but is subject to change. Users are advised to check with the authors before proceeding. The permfile AEG-LIB/SC-LB is the AEG Scientific Library, from which MLPKG is drawn. Files 01, 02, 03, 39 and 40 are internal scratch files. File 04, assigned to output tape, will contain calculated data from tasks PREJET, JETMIX, GMOD, and SPALDG, and may be used for input to a subsequent PLUMOD run (as file 07) to re-start JETMIX or recompute SPALDG or EMIS with new parameters.

3.2 File Storage and Input Data Format

The PLUMOD system consists of six main calculation programs which are designated as follows:

Program Name	Function
PREJET	Problem input and initialization.
JETMIX	Free mixing calculation for the plume.
GMOD	Modification of initial "g" profiles.
SPALDG	Spalding "g" function calculation for the plume.
EMIS	Streamtube mixing. Chemical kinetics. Emission output.
CFILE	Merge or decollate output/input data stored on binary files.

The first five programs are normally run in a sequential fashion with output data passed to succeeding programs via binary output files. The program CFILE is used to merge the data produced by PREJET, JETMIX, and SPALDG for storage on a single file (usually magnetic tape) or to restore the merged data on the original files. The transfer files utilized in the PLUMOD system are:

File Code	Data Generated by:	Date Used by:
1	PREJET	JETMIX, GMOD, EMIS
2	JETMIX	SPALDG, EMIS
3	SPALDG	EMIS
4	CFILE	Output file with merged data from files 1, 2, 3
7	CFILE	Input file with merged data for restoration of files 1, 2, 3

As indicated above, the programs are normally run in a sequential fashion. The job may be terminated, however, after any program and the data saved on file 4 for a subsequent restart by running program CFILE. Also, CFILE may be initially run to restore data for either a restart (JETMIX only) or a continuation of the total PLUMOD job from the beginning of any program.

Card data are input to the PLUMOD system on file code I* (5) and consist of both formatted and Namelist data. The first three data cards are used for case identification and are read with a 1X,10A6 format by MAINP; viz,

▽2

(Name - up to 60 alphanumeric characters)

(Address - up to 60 alphanumeric characters)

(Ident. - up to 60 alphanumeric characters)

The problem input to the system follows these cards and consists of a program card designating the program name and the existence of an input file and/or an output file. The general format of the program card is as follows:

	(input file ?)	(output file?)
▽2	▽12	▽14
(Program name)	(T or F)	(T or F)

In all cases, the program card is followed by one (1) or more Namelist sets which are read by the calculation programs. A typical input card deck to run the entire PLUMOD system and save an output file is shown below:

▽2	▽12	▽14
PREJET	F	T
$TSTDAT	(Namelist data for READIN) $	
$A	(Namelist data for JETPRF) $	
JETMIX	T	T
$A	(Namelist data for JETINP) $	
GMOD	T	T
$A	(Namelist data for GMOD) $	
SPALDG	T	T
$CHANGE	(Namelist data for SPALD) $	
EMIS	T	F
$A	(Namelist data for MXKINO) $	
$INPUT	(Namelist data for MFMAIN) $	
$PROB	(Namelist data for KINPP, if GCKP is used) $	
CFILE	T	T
$A	(Namelist data for CFILE) $	

Note that the card input to CFILE may follow any of the individual input sets if it desired to only run a portion of the problem and save the cumulative output on file 4. Also, if the problem is to be restarted (JETMIX) or continued, the CFILE input must precede the card input to the program(s) to be executed. Since the PLUMOD system normally requires a large amount of processing time and utilizes a large amount of computer resources, no provision exists for the execution of multiple cases.

Detailed descriptions of the variables in the various Namelists are given in the table below. Here, BITS means the Honeywell standard noise word (0377 777 777 777). The "initial value" is the value assigned to the variable by the program prior to reading the input. The "default value" is the value given the variable after reading input in the event the initial value is not changed. In cases where internal initialization is not provided, recommended values are given where appropriate. For rules concerning input of Namelist data, the reader is referred to the Honeywell Fortran IV manual (Reference 8)

NAMELIST / TSTDAT / Engine test data read by READIN

Variable Name	Type	Dim.	Init. Value	Default Value	Description
PO	R	1	14.696	-	Ambient air static pressure, psia.
TO	R	1	518.69	-	Ambient air static temperature, ° R
HUM	R	1	.00531	-	Air specific humidity, lb moisture/lb dry air
VO	R	1	0.0	-	Flight speed, ft/sec
RADJ	R	1	1.0	-	Outer radius of exhaust jet, ft
AMBT	R	5			Equivalent to PO thru RADJ
T2	R	1	518.69	-	Ram air total temperature, ° R
AR5	R	1	0.02	-	Turbine exit fuel-air ratio, lb fuel/lb dry air
EINO2C	R	1	20.0	-	Main combustor NOX emission index, lb NO_2/Klb fuel
BETA	R	1	0.0	-	Engine bypass ratio, lb fan air/lb core air
T25	R	1	518.69	-	Fan discharge temperature, ° R
CYCLE	R	5			Equivalent to T2 thru T25
FUEL	R	3	2.0	-	Hydrogen-carbon atom ratio in fuel
			537.0	-	Fuel temperature, ° R
			BITS	Estimated	Lower heating value of fuel, B

NAMELIST / TSTDAT / (Continued)

Variable Name	Type	Dim.	Init. Value	Default Value	Description
CAROL	R	4, 11	BITS	-	CAROL gas analyses of CO, CO_2, Hydrocarbon, and NO_x
RADII	R	11	BITS	-	Radial locations of probe measurements
PT	R	11	BITS	-	Probe impact pressure measurements, psia
PS	R	11	BITS	PO	Static pressure at probe locations, psia
BLOC	R	11	BITS	BETA	Local bypass ratio at probe locations
EICOC	R	11	BITS	0.0	Main combustor CO Emission Index, lb CO/klb fuel
ALDAT	R	10, 11	BITS		Alternate vector for CAROL thru EICOC info. If both are given, CAROL etc replace values in ALDAT. Example: PT (n), if given, replaces ALDAT (6, n). ALDAT (10, n) not used.
TITLE	H	20	Blank	-	Output page heading information
SF	R	1	1.0	-	Scale factor for CAROL data (leave 1.0 if analysis given in mole fractions, set 1E-6 if in parts per million, etc)
PRINT	R	30	BITS	0.1*RADJ 10*RADJ BITS	Axial stations (feet) at which output is desired

NAMELIST / A / JETMIX parameters and controls read by JETPRF

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
TIJET	R	1	-	0.02	Initial jet turbulence intensity (u_p'/u_p)
TIE	R	1	-	0.0	External (ambient) turbulence intensity
RG	R	1	-	53.34	Gas constant (ft lbf/lbm ° R)
PR	R	1	-	0.72	Laminer Prandtl number
PRT	R	1	-	0.70	Turbulent Prandtl number
SC	R	1	-	201.6	Sutherland constant, ° R
TREF	R	1	-	0.0	Reference temperature for viscosity, ° R
MUREF	R	1	-	0.0	Reference viscosity, lbm/ft sec
NM	I	1	-	95	Maximum number of JETMIX streamlines before mesh redistribution occurs
CA	R	1	-	6.4303	For jet dummy gases *
CB	R	1	-	8.929E-4	For jet dummy gases *
CC	R	1	-	5.989E-8	For jet dummy gases *
CAA	R	1	-	6.4303	For ambient air *
CBA	R	1	-	8.929E-4	For ambient air *
CCA	R	1	-	5.989E-8	For ambient air *

*Coefficients for determination of specific heat as a quadratic function of local static temperature (° R). $cp = a + bT = cT^2$

NAMELIST / A / (JETPRP - Continued)

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
CT1	R	1	-	0.23	Constants in the empirical equation for turbulence length scale in the potential core region
CT2	R	1	-	0.38	
CT3	R	1	-	0.23	Constants in the empirical equation for turbulence length scale in the transition region
CT4	R	1	-	0.05	
CT5	R	1	-	0.38	
CT6	R	1	-	1.4	
CT7	R	1	-	0.43	
CT8	R	1	-	0.1875	Constant in the empirical equation for turbulence length scale in the fully developed region

NAMELIST / A / JETMIX parameters and controls read by JETINP

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
DIAJ	R	1	-	*	Diameter (AXI=T) or 2* half height of primary jet, in
MJET	R	1	BJTS	*	Primary jet Mach number
TJET	R	1	518.688	*	Primary jet static temperature, ° R
PTJET	R	1	BITS	*	Primary jet stagnation pressure, psia
VJET	R	1	-	*	Primary jet velocity, fps (u_J)
TIJET	R	1	0.	*	Primary jet turbulence intensity (Referenced to YJET)
PE	R	1	14.69594	*	Ambient pressure, psia
VE	R	1	-	*	External stream velocity, fps
ME	R	1	-	*	External stream Mach number
TIE	R	1	0.	*	External stream turbulence intensity (Referenced to VJET)
TE	R	1	518.688	*	Ambient temperature, ° R
AXI	L	1	T	-	T = axisymmetric, F= plane (2 - D)
NJ	I	1	-	*	Number of mesh points in primary jet initial profile
NM	I	1	95	*	Maxumum number of mesh points
GAM	R	1	-	-	Not used

NAMELIST / A / (JETINP - Continued)

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
RG	R	1	53.34 (air)	*	Gas constant, ft lb_f/ lbm ° R
PR	R	1	0.72 (air)	*	Prandtl number
PRT	R	1	1.0	*	Turbulent Prandtl number
X	R	100	-	**	Dimensionless stream-wise coordinate (free mixing) $X = x/d_J$
XPRN	I	100	0	2.0	Profile print indicator ≥ 1 = Print, 0 = Do not print
SC	R	1	BITS	*	Sutherland constant for viscosity calculation, ° R. If not input, air values are used.
TREF	R	1	BITS	*	Reference temperature for viscosity calculation, ° R
MUREF	R	1	BITS	*	Reference viscosity, lbm/ft sec
SP	R	1	-	-	Not used
SV	R	1	-	-	Not used
SLEN	R	1	-	-	Not used
DPRIN	L	1	F	-	T - print profiles at each step
PLOT	L	1	-	-	Not used
NRED	I	1	-	-	Not used
PM	R	10	-	-	Not used
CXPC	R	1	.02	0.15**	Step size control - potential core region. Fraction % of mixing zone width

NAMELIST / A / (JETINP - Continued)

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
CXTP	R	1	.02	.015	Step size control - transition/similar region. Fractional % of jet radius or half height.
MCHANG	L	1	-	-	Not used
IDENT	H	10	Blank	*	Identification block
Y	R	200	-	*	Dimensionless normal coordinate $Y = y/r_J$
UD	R	200		*	Dimensionless velocity $UD = u/u_J$
THD	R	200	-	*	Dimensionless temperature, $THD = T/T_J$
ED	R	200	-	*	Dimensionless turbulence energy, $ED = e/e_J$
TID	R	200	-	*	Turbulence intensity (Reference to VJET)
CT1	R	1	-	*	Turbulence intensity
CT2	R	1	-	*	↓
CT3	R	1	-	*	↓
CT4	R	1	-	*	↓
CT5	R	1	-	*	(See input to JETPRF)
CT6	R	1	-	*	↓
CT7	R	1	-	*	↓
CT8	R	1	-	*	↓
CT9	R	1	-	-	Not used
CTM	R	1	.23	-	Coannular mixing, turbulence scale proportionality constant for merged region

NAMELIST / A / (JETINP - Continued)

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
CTS	R	1	.23	-	Coannular mixing, turbulence scale proportionality constant for secondary jet
CTP	R	1	.175	-	Coannular mixing, turbulence scale proportionality constant for primary jet
B	R	500	-	-	Not used
NB	I	1	-	-	Not used
TAB	R	5	-	-	Not used
D	R	800	-	-	Not used
ND	I	1	-	-	Not used
TAD	R	4	-	-	Not used
RESTRT	R	1	BITS	**	X station for restart of problem
CK	R	1	1.06064475	1.0466475	Mesh parameters for transition, similar region
DY1	R	1	.001	.002	$\psi_i = \frac{DY1((CK)^{i-1}-1)}{(CK-1)}$ i = 1,NMSH
NMSH	I	1	71	-	
MIX	L	1	F	-	Not used
MAXIT	I	1	-	-	Not used
CF	R	1	-	-	Not used
TOL	R	1	-	-	Not used
SUPB	L	1	-	-	Not used
RD	R	100	-	-	Not used
XD	R	100	-	-	Not used
YCB	R	100	-	-	Not used

NAMELIST / A / (JETINP - Continued)

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
TWO	L	1	F	-	Jet configuration - F - single mixing region T - coannular mixing region
DIAO	R	1	-	-	Diameter (AXI = T) or 2* half height of secondary jet, in.
MJETO	R	1	BITS	-	Secondary jet Mach number
TJETO	R	1	518.688	-	Secondary jet static temperature, ° R
VJETO	R	1	-	-	Secondary jet velocity, fps
PTJETO	R	1	BITS	-	Secondary jet stagnation pressure, psia
TIJETO	R	1	0.	-	Secondary jet turbulence intensity (Relative to VJET)
NJO	I	1	-	-	Mesh index of outermost point in secondary jet
NC	I	1	12	*	Number of active constituents
CNAME	H	12	12	*	Names of active constituents
ALJ	R	12	-	*	Primary jet stream species mole fractions
ALJO	R	12	-	-	Secondary jet stream species mole fractions
ALE	R	12	-	*	External (boundary) species mole fractions
SCM	R	12	.7	*	Effective Schmidt number for individual species

NAMELIST / A / (JETINP - Continued)

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
TCPRF	R	-	-	-	Not used
HCPRF	R	-	-	-	Not used
CPC	R	(3, 12)	-	*	Coefficients in polynominal representation of species molar heat capacities
ALX	R	(100, 12)	-	*	Species mole fraction profiles
DIFF	L	1	T	-	Species diffusion - F - no, T - yes
CSC	R	1	1000.	-	Scaling constant for file storage
NDIGIT	I	1	-	-	Not used
LTERP	L	1	T	-	Transition region scale interpolation T - linear F - exponential
CDIFF	-	1	-	-	Not used
CHI	R	1	.586	*	Diffusion parameter (turbulence energy equation)
SCLD	L	1	F	T	T - Use ambient gas mole fraction to determine inner edge of mixing zone F - Use velocity to determine inner edge of mixing zone
ALXLIM	R	1	.0001	-	Limiting ambient gas mole fraction used to locate inner edge of mixing zone

NAMELIST / A / (JETINP - Continued)

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
CJRMOD	R	1	-	1.0	Fraction of jet radius used as reference length in determining mixing scale within potential core

* Transmitted from JETPRF

** See Section 3.4

NAMELIST / A / Specifications read by GMOD

Variable Name	Type	Dim.	Init. Value	Default Value	Description
PDUM	R	20	-	-	If PDUM (1) ≠ 0., modified profile parameters are printed
YAIR	R	1	BITS	0.0	Mass fraction of ambient air regarded as "hot" gas
YMIN	R	1	0.30	-	Minimum mass fraction hot gas at any point

NAMELIST / CHANGE / Parameters and controls read by SPALD

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
ALFA	R	1	-	0.2	Coefficients in Spalding heterogeneity conservation equation (Reference 1, Section 4.4)
CG1	R	1	-	2.7	
CG2	R	1	-	0.67	
SIGMAG	R	1	-	0.7	
THETA	R	1	-	0.5	Weighting factor for finite-difference solution
DX	R	1	1000.	0.01	Initial axial step length, jet diameters
DXPRN	R	1	-	1.0	Not used
XSTOP	R	1	1000.	24.0	Last axial station (x/d_j) at which heterogeneity parameter will be computed
NSZ1	I	1	-	25	Number of equally spaced streamlines at initial axial station
PRINT	L	1	F	-	Diagnostic profiles of G are printed

NAMELIST / A / Control variables read by MXKINO

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
PDUM	R	20	0.0	-	Diagnostic printout controls for KINET, EQKIN, MFMAIN, READT, MXFLUT, SCKP, HYCARB, GCKP
CHEMK	I	1	1	2	1 means use GCKP; 2 means use SCKP
PSTA	R	30	BITS	-	Axial stations (jet diameters) at which output is desired. Overrides values in $TSTDAT.
NIST	I	1	2	-	Not used
COREQ	L	1	-	T	Use current rate equation for CO consumption reaction in SCKP

NAMELIST / INPUT / Control variables read by MFMAIN

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
X	R	1	0.0	-	Do not use
XL	R	1	1.2	-	Do not use
NTUBES	I	1	-	-	Do not use
DX	R	1	.05	-	Mixing step size, jet diameters
ALFLIM	R	1	-	-	Inoperable
CMIXST	R	1	2.0	-	Factor for automatic increase in DX
NIST	I	1	10	-	Number of mixing steps per reaction step

NAMELIST / PROB / GCKP Control variables used by KINPP (Include only if MXKINO $A CHEMK = 1)

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
HMIN	R	1	0.5E-7	1.3E-8	Minimum integration step size, sec.
HMAX	R	1	0.5E-4	1E-4	Maximum step size, sec.
HINT	R	1	0.0	2.6E-8	Initial step size, sec.
EMAX	R	1	0.0001	0.01	Maximum projected relative error, used for internal step size adjustment
ALLM1	L	1	F	F	All third-body collision effectivenesses are 1.0
ELIM	L	1	F	-	Not used
CONC	L	1	T	-	" "
EXCHR	L	1	F	-	" "
IPRCOD	I	1	3	-	" "
ITPSZ	I	1	5	-	" "
XTB	R	40	0.0	-	" "
ATB	R	40	0.0	-	" "
NTB	I	1	0	-	" "
CX3	R	1	0.0	-	" "
CX2	R	1	0.0	-	" "
CX1	R	1	0.0	-	" "
CXO	R	1	0.0	-	" "
LSUBM	R	1	0.0	-	" "

NAMELIST / PROB / (Continued)

Variable Name	Type	Dim.	Init. Value	Recomm. Value	Description
ETA					Not used
D	R	1	0.0	-	" "
VISC	R	1	0.0	-	" "
BETA	R	1	0.0	-	" "
END	R	1	0.0	-	" "
DELP	R	1	0.0	-	" "
PRINT	R	50	-	-	" "
NPRNTS	I	1	0	-	" "
APRINT	R	50	0.0	-	" "
EVSTEP	L	1	F	-	" "
DBUGO	L	1	F	-	" "
COMBUS	L	1	F	-	" "
SHOCK	L	1	F	-	" "
TCON	L	1	F	-	" "
RHOCON	L	1	F	-	" "
BRIEF	L	1	F	-	" "
TIMLMT	R	1	25.0	-	" "

NAMELIST / A / File manipulation input read by CFILE

Variable Name	Type	Dim.	Init. Value	Default Value	Description
MERGE	L	1	F	-	T = Merge files (store on file code 4)
SPLIT	L	1	F	-	T = Restore files (restore files stored on file code 7)
NF	I	1	0	-	File(s) present indicator: =1 PREJET data =2 PREJET & JETMIX data =3 PREJET, JETMIX, SPALDG data

3.3 Sample Calculation

The following input data cards, representing data from a test of a J85-5 engine at mid AB power setting (Reference 1), were input to the plume model:

```
NAME= D A V E   F E R G U S O N
ADDRESS=   EVENDALE OHIO
IDENT= TEST CASE FINAL
PREJET    F 1
$YSTDAT
TITLE=12=J85-5 MID,AB
TITLE(9)=23HEPTC DEMONSTRATION TEST
TITLE(19)=5-RUN 32-3
A=AI=13.0,515,.0027,0.,675,
CYCLE=515,.01776=,4.6,0,515,
FUEL=2,552,
ALDATA=
 3,30219E-03, 6,23091E-02, 3.00073E-03, 4.93947E-05,                A200.089
+0.43734E-02, 2.50000F+01,                 ....                      B200.089
 2,00761E-03, 7.14251E-02, 1.40717E-03, 5.71395E-05,                 A100.088
+1.37310E-01, 2.91500F+01.                 ....                      B100.088
 2.01767E-03, 7.91045E-02, 4,69957E-04, 7.00475E-05,                 A200.087
+2.03277E-01, 2.74590E+01,                 ....                      B200.087
 1.32004E-03, 9.14631E-02, 3.04208E-05, 8.16647E-05,                 A200.094
 1.79695E-01. 2.91203F+01,                 ....                      B200.094
 1,24900E-03, 8.96240F-02, 1.56321E-05, 7.90831E-05,                 A200.095
 4.99455E-01, 2.91203E+01,                 ....                      B200.095
 1.05795E-03, 7.70092E-02, 2,34390E-04, 5.77418E-05,                 A200.083
+5.33242E-01, 2.72900F+01,                 ....                      B200.083
 2.02814E-03, 6.77975E-02, 4.64957E-04, 5.41270E-05,                 A100.083
+5.51031E-01, 2.73400F+01,                 ....                      B100.083
 3.09655E-03, 6.12544F-02, 9.99908E-04, 4.59525E-05,                 A200.097
 4.20109E-01, 2.54400E+01,                 ....                      B200.097
 2.19759E-03, 2.12984F-02. 1.00606E-03, 1.88457E-05,                 A100.097
 6.41544E-01, 1.70199E+01.                 ....                      B100.097
 2.05429E-03, 2.25290E-02. 1,08606E-03, 1.07062E-05,                 A200.097
 6.64400E-01, 1.56100E+01.                 ....                      B200.097
 1,15955E-03, 1.30754F-02, 7.05529E-04, 1.29889E-05,                 A100.082
+0.73404E-01, 1.50000E+01.                 ....,                     B100.082
EICNC=33.9,
PRINT=.0675,3.75,7.5,30,
$END J85 MID AB

$A
TIJET=.02,
TIE=0.,
H=53.34,P4=,72,PR7=,70,
SC=201.6,TREF=0.,MUREF=0.,
N==5,
CA=6.6303,CA=8.929E-4,CC=5.909E-8,
CAA=6.6303,CBA=4.929E-4,CCA=5.909E-8,
CT1=.23,CT2=,38,CT3=.23,
CT4=.05,
CT5=.38,CT6=1.4,CT7=.43,
CT9=.1475,
$
JETMIX     T T
$A
XP4N=2,,
DV1=,002,CK=1.0466475,
PCLOT,
CAPC=0.15,CXT=.015,
CJ4=20=1.0,
X=0.,.02,.05,.1,.2,.5,1.,1.5,2.,2.5,3.,3.5,4.,4.5,5.,5.5,6.,6.5,7.,
7.5,8.,9.,10,,11.,12.,13.,15.,17.,19.,21.,23.,25.,
X(11)=2.7777778,X(16)=5.5555556,X(24)=11.111111,X(31)=22.222222,       J85 MID
$
GMJD       T T
```

```
$A POUM=1,YAIR=0.0,YMIN=0.30$
$PALDG     T T
$CHANGE
ALFA=.2,CG1=2.7,CG2=.134,SIGMAG=.7,THETA=.5,DX=.01,DXPRN=1.,
PRINT=T,
XSTOP=24.,NSZ1=25,
CG2=0.07,
$
EMIS       T T
$A
POUM(20)=39,
CHEMK=2,
CORFST=T,
PSTAR=.05,.5,1.0,2.0,2.7777778,5.5555556,11.111111,22.222222,          JBS MID
$
$INPUT DX=.05,NIST=10$
CFILE      T T
$A
MERGE=T,NF=3,
$
```

Using the input data given above, PLUMOD generated the computation reports shown below. For brevity, JETMIX reports at the first and last computation stations only are reproduced here. The SPALDG reports at only the initial stations are shown, and some EMIS reports have been omitted.

* FREE JET PROGRAM *

* AXISYMMETRIC * * NONISOTHERMAL * * COMPRESSIBLE * JET

** SINGLE MIXING REGION **

NAME= D A V E F E R G U S O N

ADDRESS= EVENDALE OHIO

* INPUT AND INITIAL CONDITIONS *

EXTERNAL CONDITIONS		JET DISCHARGE PARAMETERS		GAS PROPERTIES	
TE =	515.000	DIAJ =	16.20000	GAM =	1.40000
PE =	1.3630E 01	MJET =	1.0989	RG =	53.34000
VE =	0.	TJET =	2330.982	PR =	0.72000
ME =	0.	PTJET=	2.8014E 01	PRT =	0.70000
TIE=	0.	VJET =	2503.983	SC =	
		TIJET=	2.0000E-02	TREF =	
		FLOWJ=	4.4609E 00	MUREF=	

* INITIAL PROFILES *

N	Y	PSI	U	THETA	TI	GAS1	GAS2	GAS3	GAS4	GAS5	GAS6
1	0.	0.	0.92308828	0.88319692	2.0000000E-02	1.0000E 00	0.	0.	0.	0.	0.
2	1.0000E-02	4.1247E-07	0.92308828	0.88319692	2.0000000E-02	1.0000E 00	0.	0.	0.	0.	0.
3	7.5596E-02	2.3565E-05	0.92308828	0.88319692	2.0000000E-02	1.0000E 00	0.	0.	0.	0.	0.
4	1.4117E-01	8.2203E-05	0.92308828	0.88319692	2.0000000E-02	1.0000E 00	0.	0.	0.	0.	0.
5	1.4617E-01	8.8135E-05	0.93226640	0.89063979	2.0000000E-02	9.0000E-01	1.0000E-01	0.	0.	0.	0.
6	1.5117E-01	9.4293E-05	0.96597890	0.92041126	2.0000000E-02	5.0000E-01	5.0000E-01	0.	0.	0.	0.
7	1.5617E-01	1.0069E-04	1.00569138	0.95018272	2.0000000E-02	1.0000E-01	9.0000E-01	0.	0.	0.	0.
8	1.6117E-01	1.0733E-04	1.01486951	0.95762559	2.0000000E-02	0.	1.0000E 00	0.	0.	0.	0.
9	2.2972E-01	2.1940E-04	1.01486951	0.95762558	2.0000000E-02	0.	1.0000E 00	0.	0.	0.	0.
10	2.9828E-01	3.7078E-04	1.01486951	0.95762558	2.0000000E-02	0.	1.0000E 00	0.	0.	0.	0.
11	3.0328E-01	3.8336E-04	1.02151354	0.96435182	2.0000000E-02	0.	9.0000E-01	1.0000E-01	0.	0.	0.
12	3.0828E-01	3.9613E-04	1.04808968	0.99125671	2.0000000E-02	0.	5.0000E-01	5.0000E-01	0.	0.	0.
13	3.1328E-01	4.0909E-04	1.07466582	1.01816159	2.0000000E-02	0.	1.0000E-01	9.0000E-01	0.	0.	0.
14	3.1828E-01	4.2224E-04	1.08130984	1.02488783	2.0000000E-02	0.	0.	1.0000E 00	0.	0.	0.
15	3.9525E-01	6.5093E-04	1.08130984	1.02488783	2.0000000E-02	0.	0.	1.0000E 00	0.	0.	0.
16	4.7223E-01	9.2597E-04	1.08130984	1.02488783	2.0000000E-02	0.	0.	1.0000E 00	0.	0.	0.
17	4.7723E-01	9.4872E-04	1.08917648	1.03446654	2.0000000E-02	0.	0.	9.0000E-01	1.0000E-01	0.	0.
18	4.8223E-01	9.6557E-04	1.12064299	1.07278134	2.0000000E-02	0.	0.	5.0000E-01	5.0000E-01	0.	0.
19	4.8723E-01	9.8848E-04	1.15210952	1.11109614	2.0000000E-02	0.	0.	1.0000E-01	9.0000E-01	0.	0.
20	4.9223E-01	1.0085E-03	1.15997616	1.12067485	2.0000000E-02	0.	0.	0.	1.0000E 00	0.	0.
21	5.6990E-01	1.3455E-03	1.15997616	1.12067485	2.0000000E-02	0.	0.	0.	1.0000E 00	0.	0.
22	6.4758E-01	1.7319E-03	1.15997616	1.12067485	2.0000000E-02	0.	0.	0.	1.0000E 00	0.	0.
23	6.5258E-01	1.7583E-03	1.15847759	1.11823747	2.0000000E-02	0.	0.	0.	9.0000E-01	1.0000E-01	0.
24	6.5758E-01	1.7852E-03	1.15249342	1.10848793	2.0000000E-02	0.	0.	0.	5.0000E-01	5.0000E-01	0.
25	6.6258E-01	1.8123E-03	1.14649925	1.09873840	2.0000000E-02	0.	0.	0.	1.0000E-01	9.0000E-01	0.
26	6.6758E-01	1.8397E-03	1.14499070	1.09630102	2.0000000E-02	0.	0.	0.	0.	1.0000E 00	0.
27	7.1047E-01	2.0814E-03	1.14499070	1.09630102	2.0000000E-02	0.	0.	0.	0.	1.0000E 00	0.
28	7.5337E-01	2.3422E-03	1.14499070	1.09630102	2.0000000E-02	0.	0.	0.	0.	1.0000E 00	0.
29	7.5837E-01	2.3733E-03	1.13501762	1.08825634	2.0000000E-02	0.	0.	0.	0.	9.0000E-01	1.0000E-01

30	7.6337E-01	2.4046E-03	1.09512530	1.05607767	2.0000000E-02	0.	0.	0.	0.	5.0000E-01	5.0000E-01
31	7.6837E-01	2.4358E-03	1.05523300	1.02389900	2.0000000E-02	0.	0.	0.	0.	1.0000E-01	9.0000E-01
32	7.7337E-01	2.4571E-03	1.04525992	1.01585433	2.0000000E-02	0.	0.	0.	0.	0.	1.0000E 00
33	7.8245E-01	2.5245E-03	1.04525992	1.01585433	2.0000000E-02	0.	0.	0.	0.	0.	1.0000E 00
34	7.9152E-01	2.5825E-03	1.04525992	1.01585433	2.0000000E-02	0.	0.	0.	0.	0.	1.0000E 00
35	7.9652E-01	2.6147E-03	1.03838681	1.00910872	2.0000000E-02	0.	0.	0.	0.	0.	9.0000E-01
36	8.0152E-01	2.6472E-03	1.01089436	0.98212629	2.0000000E-02	0.	0.	0.	0.	0.	5.0000E-01
37	8.0652E-01	2.6798E-03	0.98340190	0.95514389	2.0000000E-02	0.	0.	0.	0.	0.	1.0000E-01
38	8.1152E-01	2.7127E-03	0.97652879	0.94839824	2.0000000E-02	0.	0.	0.	0.	0.	0.
39	8.3550E-01	2.8732E-03	0.97652879	0.94839824	2.0000000E-02	0.	0.	0.	0.	0.	0.
40	8.5947E-01	3.0383E-03	0.97652879	0.94839524	2.0000000E-02	0.	0.	0.	0.	0.	0.
41	8.6447E-01	3.0733E-03	0.96736145	0.94163066	2.0000000E-02	0.	0.	0.	0.	0.	0.
42	8.6947E-01	3.1082E-03	0.93069212	0.91456032	2.0000000E-02	0.	0.	0.	0.	0.	0.
43	8.7447E-01	3.1431E-03	0.89402278	0.88748998	2.0000000E-02	0.	0.	0.	0.	0.	0.
44	8.7947E-01	3.1779E-03	0.88485544	0.88072240	2.0000000E-02	0.	0.	0.	0.	0.	0.
45	9.0211E-01	3.3378E-03	0.88485544	0.88072240	2.0000000E-02	0.	0.	0.	0.	0.	0.
46	9.2975E-01	3.5018E-03	0.88485544	0.88072240	2.0000000E-02	0.	0.	0.	0.	0.	0.
47	9.2975E-01	3.5382E-03	0.83588435	0.84692842	2.0000000E-02	0.	0.	0.	0.	0.	0.
48	9.3475E-01	3.5729E-03	0.69000000	0.71175249	2.0000000E-02	0.	0.	0.	0.	0.	0.
49	9.3975E-01	3.6138E-03	0.44411563	0.57657656	2.0000000E-02	0.	0.	0.	0.	0.	0.
50	9.4475E-01	3.6317E-03	0.39514454	0.54278257	2.0000000E-02	0.	0.	0.	0.	0.	0.
51	9.5116E-01	3.6666E-03	0.39514454	0.54278257	2.0000000E-02	0.	0.	0.	0.	0.	0.
52	9.5758E-01	3.7015E-03	0.39514454	0.54278257	2.0000000E-02	0.	0.	0.	0.	0.	0.
53	9.6258E-01	3.7291E-03	0.38781301	0.54272576	2.0000000E-02	0.	0.	0.	0.	0.	0.
54	9.6758E-01	3.7553E-03	0.35848690	0.54249849	2.0000000E-02	0.	0.	0.	0.	0.	0.
55	9.7258E-01	3.7796E-03	0.32916079	0.54227122	2.0000000E-02	0.	0.	0.	0.	0.	0.
56	9.7758E-01	3.8027E-03	0.32182926	0.54221440	2.0000000E-02	0.	0.	0.	0.	0.	0.
57	9.7821E-01	3.8056E-03	0.32182926	0.54221440	2.0000000E-02	0.	0.	0.	0.	0.	0.
58	9.7885E-01	3.8085E-03	0.32182926	0.54221440	2.0000000E-02	0.	0.	0.	0.	0.	0.
59	9.8385E-01	3.8314E-03	0.31281132	0.53161137	2.0000000E-02	0.	0.	0.	0.	0.	0.
60	9.8885E-01	3.8538E-03	0.27673957	0.48917927	2.0000000E-02	0.	0.	0.	0.	0.	0.
61	9.9024E-01	3.8598E-03	0.24066781	0.44678716	2.0000000E-02	0.	0.	0.	0.	0.	0.
62	9.9169E-01	3.8657E-03	0.23164987	0.43618419	2.0000000E-02	0.	0.	0.	0.	0.	0.
63	9.9442E-01	3.8773E-03	0.23164987	0.43618914	2.0000000E-02	0.	0.	0.	0.	0.	0.
64	9.9723E-01	3.8889E-03	0.23164987	0.43618419	2.0000000E-02	0.	0.	0.	0.	0.	0.
65	9.9861E-01	3.8945E-03	0.20899489	0.41465942	1.8000000E-02	0.	0.	0.	0.	0.	0.
66	1.0000E 00	3.8993E-03	0.11582494	0.32856054	1.0000000E-02	0.	0.	0.	0.	0.	0.
67	1.0050E 00	3.9041E-03	0.02316499	0.24246166	2.0000000E-03	0.	0.	0.	0.	0.	0.
68	1.0100E 00	3.9100E-03	0.	0.22093694	0.	0.	0.	0.	0.	0.	0.

* INITIAL PROFILES *

N	Y	PSI	GAS7	GAS8	GAS9	GAS10	GAS11	GAS12
1	0.	0.	0.	0.	0.	0.	0.	0.
2	1.3333E-02	4.1247E-07	0.	0.	0.	0.	0.	0.
3	7.5586E-02	2.3365E-05	0.	0.	0.	0.	0.	0.
4	1.4117E-01	8.2203E-05	0.	0.	0.	0.	0.	0.
5	1.4617E-01	8.8133E-05	0.	0.	0.	0.	0.	0.
6	1.5117E-01	9.4293E-05	0.	0.	0.	0.	0.	0.
7	1.5617E-01	1.0069E-04	0.	0.	0.	0.	0.	0.
8	1.6117E-01	1.0731E-04	0.	0.	0.	0.	0.	0.
9	2.2972E-01	2.1943E-04	0.	0.	0.	0.	0.	0.
10	2.9528E-01	3.7078E-04	0.	0.	0.	0.	0.	0.
11	3.0329E-01	3.8335E-04	0.	0.	0.	0.	0.	0.
12	3.0928E-01	3.9613E-04	0.	0.	0.	0.	0.	0.
13	3.1524E-01	4.0929E-04	0.	0.	0.	0.	0.	0.
14	3.1820E-01	4.2224E-04	0.	0.	0.	0.	0.	0.
15	[illegible]	6.5096E-04	0.	0.	0.	0.	0.	0.
16	[illegible]	9.2397E-04	0.	0.	0.	0.	0.	0.
17	4.7723E-01	9.4972E-04	0.	0.	0.	0.	0.	0.
18	4.8223E-01	9.6557E-04	0.	0.	0.	0.	0.	0.
19	4.8723E-01	9.8348E-04	0.	0.	0.	0.	0.	0.
20	4.9223E-01	1.0085E-03	0.	0.	0.	0.	0.	0.
21	5.6490E-01	1.3055E-03	0.	0.	0.	0.	0.	0.
22	6.0735E-01	1.7315E-03	0.	0.	0.	0.	0.	0.
23	6.5254E-01	1.7581E-03	0.	0.	0.	0.	0.	0.
24	6.5754E-01	1.7852E-03	0.	0.	0.	0.	0.	0.
25	6.6254E-01	1.8123E-03	0.	0.	0.	0.	0.	0.
26	6.6754E-01	1.8397E-03	0.	0.	0.	0.	0.	0.
27	7.1047E-01	2.0954E-03	0.	0.	0.	0.	0.	0.
28	7.5337E-01	2.3422E-03	0.	0.	0.	0.	0.	0.
29	7.5837E-01	2.3733E-03	0.	0.	0.	0.	0.	0.
30	7.6337E-01	2.4046E-03	0.	0.	0.	0.	0.	0.
31	7.6837E-01	2.4359E-03	0.	0.	0.	0.	0.	0.
32	7.7337E-01	2.4671E-03	0.	0.	0.	0.	0.	0.
33	7.9245E-01	2.5245E-03	0.	0.	0.	0.	0.	0.
34	7.9152E-01	2.5425E-03	0.	0.	0.	0.	0.	0.
35	7.9652E-01	2.6107E-03	1.0000E-01	0.	0.	0.	0.	0.
36	8.0152E-01	2.6472E-03	5.0000E-01	0.	0.	0.	0.	0.
37	8.0652E-01	2.6798E-03	9.0000E-01	0.	0.	0.	0.	0.
38	8.1152E-01	2.7127E-03	1.0000E 00	0.	0.	0.	0.	0.
39	8.3550E-01	2.8732E-03	1.0000E 00	0.	0.	0.	0.	0.
40	8.5947E-01	3.0193E-03	1.0000E 00	0.	0.	0.	0.	0.
41	8.6447E-01	3.0735E-03	9.0000E-01	1.0000E-01	0.	0.	0.	0.
42	8.6947E-01	3.1082E-03	5.0000E-01	5.0000E-01	0.	0.	0.	0.
43	8.7447E-01	3.1451E-03	1.0000E-01	9.0000E-01	0.	0.	0.	0.
44	8.7947E-01	3.1779E-03	0.	1.0000E 00	0.	0.	0.	0.
45	9.2211E-01	3.3579E-03	0.	1.0000E 00	0.	0.	0.	0.
46	9.2475E-01	3.5019E-03	0.	1.0000E 00	0.	0.	0.	0.
47	9.2975E-01	3.5582E-03	0.	9.0000E-01	1.0000E-01	0.	0.	0.
48	9.3475E-01	3.5729E-03	0.	5.0000E-01	5.0000E-01	0.	0.	0.
49	9.3975E-01	3.6038E-03	0.	1.0000E-01	9.0000E-01	0.	0.	0.
50	9.4475E-01	3.6317E-03	0.	0.	1.0000E 00	0.	0.	0.
51	9.5116E-01	3.6566E-03	0.	0.	1.0000E 00	0.	0.	0.
52	9.5758E-01	3.7018E-03	0.	0.	1.0000E 00	0.	0.	0.
53	9.6258E-01	3.7291E-03	0.	0.	9.0000E-01	1.0000E-01	0.	0.
54	9.6758E-01	3.7555E-03	0.	0.	5.0000E-01	5.0000E-01	0.	0.
55	9.7258E-01	3.7796E-03	0.	0.	1.0000E-01	9.0000E-01	0.	0.
56	9.7758E-01	3.8027E-03	0.	0.	0.	1.0000E 00	0.	0.
57	9.7821E-01	3.8056E-03	0.	0.	0.	1.0000E 00	0.	0.
58	9.7885E-01	3.8085E-03	0.	0.	0.	1.0000E 00	0.	0.
59	9.8395E-01	3.8314E-03	0.	0.	0.	9.0000E-01	1.0000E-01	0.

```
60  9.8885E-01   3.8538E-03 0.          0.          0.          5.0000E-01 5.0000E-01 0.
61  9.9024E-01   3.8599E-03 0.          0.          0.          1.0000E-01 9.0000E-01 0.
62  9.9164E-01   3.8657E-03 0.          0.          0.          0.         1.0000E 00 0.
63  9.9492E-01   3.8773E-03 0.          0.          0.          0.         1.0000E 00 0.
64  9.9721E-01   3.8804E-03 0.          0.          0.          0.         1.0000E 00 0.
65  9.9861E-01   3.8940E-03 0.          0.          0.          0.         9.0000E-01 1.0000E-01
66  1.0000E 00   3.8993E-03 0.          0.          0.          0.         5.0000E-01 5.0000E-01
67  1.0050E 00   3.9031E-03 0.          0.          0.          0.         1.0000E-01 9.0000E-01
68  1.0100E 00   3.9100E-03 0.          0.          0.          0.         0.         1.0000E 00
        EXP UNDERFLO  AT LOCATION      104125
        EXP UNDERFLO  AT LOCATION      102050
        EXP UNDERFLO  AT LOCATION      102050
        EXP UNDERFLO  AT LOCATION      102043
        EXP UNDERFLO  AT LOCATION      102050
        EXP UNDERFLO  AT LOCATION      102043
        EXP UNDERFLO  AT LOCATION      102047
        EXP UNDERFLO  AT LOCATION      102042
        EXP UNDERFLO  AT LOCATION      104125
        EXP UNDERFLO  AT LOCATION      104144
        EXP UNDERFLO  AT LOCATION      102050
        EXP UNDERFLO  AT LOCATION      102043
        EXP UNDERFLO  AT LOCATION      102050
        EXP UNDERFLO  AT LOCATION      102043
        EXP UNDERFLO  AT LOCATION      102050
        EXP UNDERFLO  AT LOCATION      102043
        EXP UNDERFLO  AT LOCATION      102047
        EXP UNDERFLO  AT LOCATION      102042
        EXP UNDERFLO  AT LOCATION      102047
        EXP UNDERFLO  AT LOCATION      102042
        EXP UNDERFLO  AT LOCATION      104125
        EXP UNDERFLO  AT LOCATION      104144
        EXP UNDERFLO  AT LOCATION      104144
        EXP UNDERFLO  AT LOCATION      102047
        EXP UNDERFLO  AT LOCATION      102042
        EXP UNDERFLO  AT LOCATION      102047
**THIS IS THE LAST TIME THE ABOVE MESSAGE WILL APPEAR*
```

* JET ANALYSIS PROGRAM *

PROFILES-- STA (33) X= 25.00000 PRESSURE= 13.6300

N	Y	PSI	UD	THD	TI	TTD	PTD	MACH	U	T	TOT	PTOT
			** DIMENSIONLESS **					**		** DIMENSIONAL **		
1	0.	0.	0.177637	0.375737	4.10520E-02	0.1698278	0.0638563	0.30776	444.7990	575.8355	894.5947	14.5495
2	0.22082	9.0645E-05	0.177637	0.375737	4.10520E-02	0.1698278	0.0638563	0.30776	444.7994	575.8353	894.5967	14.5495
3	0.31539	1.8552E-04	0.177271	0.375523	4.10847E-02	0.1695774	0.0636246	0.30721	443.8859	575.3361	894.0251	14.5452
4	0.39093	2.9482E-04	0.176815	0.375255	4.11292E-02	0.1692649	0.0633363	0.30652	442.7426	874.7129	893.3265	14.5412
5	0.45592	3.8675E-04	0.176307	0.374957	4.11667E-02	0.1689160	0.0630153	0.30575	441.4696	874.0169	892.5466	14.5364
6	0.51717	4.9755E-04	0.175758	0.374634	4.12111E-02	0.1685386	0.0626694	0.30492	440.0945	873.2642	891.7035	14.5314
7	0.57351	6.1139E-04	0.175173	0.374289	4.12565E-02	0.1681362	0.0623018	0.30404	438.6293	872.4611	890.8039	14.5261
8	0.62740	7.3355E-04	0.174554	0.373924	4.13026E-02	0.1677101	0.0619139	0.30310	437.0745	871.6155	889.8514	14.5206
9	0.67929	8.5524E-04	0.173902	0.373539	4.13491E-02	0.1672608	0.0615065	0.30211	435.4473	870.7134	888.8471	14.5147
10	0.72978	9.8582E-04	0.173217	0.373134	4.13957E-02	0.1667883	0.0610799	0.30108	433.7350	869.7677	887.7910	14.5084
11	0.77925	1.1224E-03	0.172500	0.372709	4.14421E-02	0.1662923	0.0606340	0.29999	431.9360	869.7769	886.6826	14.5021
12	0.82836	1.2655E-03	0.171748	0.372264	4.14880E-02	0.1657726	0.0601588	0.29855	430.0552	867.7401	885.5229	14.4955
13	0.87637	1.4151E-03	0.170963	0.371797	4.15335E-02	0.1652254	0.0596941	0.29766	428.0586	866.6525	884.3045	14.4885
14	0.92449	1.5711E-03	0.170143	0.371308	4.15775E-02	0.1646591	0.0591795	0.29641	426.0349	865.5133	883.0320	14.4812
15	0.97229	1.7359E-03	0.169286	0.370798	4.16205E-02	0.1640659	0.0586548	0.27511	423.8900	864.3225	881.7015	14.4737
16	1.02019	1.9074E-03	0.168393	0.370263	4.16620E-02	0.1634419	0.0581094	0.27374	421.6530	863.0773	880.3115	14.4659
17	1.06920	2.0870E-03	0.167461	0.369705	4.17015E-02	0.1627923	0.0575451	0.27232	419.3205	861.7763	878.8595	14.4577
18	1.11645	2.2750E-03	0.166491	0.369122	4.17388E-02	0.1621181	0.0569555	0.27084	416.8596	860.4173	877.3437	14.4492
19	1.15501	2.4719E-03	0.165479	0.368513	4.17734E-02	0.1614062	0.0563460	0.25930	414.3573	858.9983	875.7615	14.4405
20	1.21397	2.6777E-03	0.164426	0.367878	4.18050E-02	0.1606677	0.0557193	0.25769	411.7203	857.5168	874.1159	14.4314
21	1.25542	2.8935E-03	0.163330	0.367215	4.18331E-02	0.1598973	0.0550600	0.25601	408.9753	855.9757	872.3584	14.4220
22	1.31343	3.1189E-03	0.162189	0.366523	4.18573E-02	0.1590939	0.0543826	0.28426	406.1189	854.3570	870.5931	14.4122
23	1.36436	3.3550E-03	0.161002	0.365801	4.18771E-02	0.1582563	0.0536917	0.28244	403.1474	852.6744	868.7211	14.4021
24	1.41540	3.6021E-03	0.159768	0.365047	4.18920E-02	0.1573832	0.0529568	0.28057	400.0573	850.9188	866.7595	14.3917
25	1.46750	3.8609E-03	0.158485	0.364262	4.19013E-02	0.1564752	0.0522077	0.27857	396.8445	849.0579	864.7357	14.3809
26	1.52244	4.1316E-03	0.157152	0.363443	4.19046E-02	0.1555250	0.0514339	0.27652	393.5054	847.1756	862.6165	14.3695
27	1.57427	4.4147E-03	0.155766	0.362589	4.19011E-02	0.1545371	0.0506350	0.27438	390.0359	845.1879	860.4082	14.3583
28	1.62907	4.7115E-03	0.154327	0.361698	4.18901E-02	0.1535079	0.0498108	0.27215	386.4316	843.1125	858.1079	14.3455
29	1.68490	5.0220E-03	0.152852	0.360770	4.18711E-02	0.1524360	0.0489609	0.26984	382.6886	840.9491	855.7123	14.3342
30	1.74193	5.3469E-03	0.151280	0.359803	4.18431E-02	0.1513196	0.0480551	0.26743	378.8025	838.6940	853.2165	14.3216
31	1.79993	5.6869E-03	0.149669	0.358795	4.18054E-02	0.1501571	0.0471831	0.26493	374.7687	836.3435	850.6184	14.3087
32	1.85926	6.0429E-03	0.147997	0.357799	4.17571E-02	0.1489466	0.0462547	0.26233	370.5828	833.8938	847.9128	14.2953
33	1.91993	6.4154E-03	0.146263	0.356648	4.16974E-02	0.1476863	0.0452997	0.25952	366.2401	831.3408	845.0959	14.2814
34	1.98193	6.8055E-03	0.144464	0.355507	4.16252E-02	0.1463742	0.0443182	0.25681	361.7357	828.6802	842.1632	14.2673
35	2.04592	7.2134E-03	0.142599	0.354317	4.15394E-02	0.1450083	0.0433099	0.25389	357.0649	825.9074	839.1104	14.2530
36	2.11036	7.6405E-03	0.140665	0.353078	4.14391E-02	0.1435865	0.0422750	0.25056	352.2227	823.0179	835.9324	14.2381
37	2.17713	8.0875E-03	0.138661	0.351756	4.13231E-02	0.1421065	0.0412136	0.24770	347.2038	820.0066	832.6245	14.2226
38	2.24553	8.5555E-03	0.136584	0.350440	4.11902E-02	0.1405661	0.0401257	0.24443	342.0032	816.8684	829.1614	14.2072
39	2.31575	9.0452E-03	0.134432	0.349036	4.10390E-02	0.1389626	0.0390116	0.24103	336.6154	813.5976	825.5974	14.1911
40	2.39790	9.5578E-03	0.132203	0.347574	4.08683E-02	0.1372936	0.0378716	0.23750	331.0351	810.1587	821.8670	14.1737
41	2.46209	1.0094E-02	0.129896	0.346050	4.04766E-02	0.1355563	0.0367063	0.23383	325.2567	806.6354	817.9841	14.1580
42	2.53842	1.0655E-02	0.127507	0.344461	4.04623E-02	0.1337480	0.0355159	0.23002	319.2740	802.9319	813.9423	14.1411
43	2.61703	1.1244E-02	0.125034	0.342802	4.02239E-02	0.1318750	0.0343011	0.22596	313.0826	799.0653	809.7559	14.1234
44	2.69937	1.1859E-02	0.122475	0.341071	3.99598E-02	0.1299071	0.0330635	0.22189	306.6751	795.0304	805.3578	14.1050
45	2.78157	1.2502E-02	0.119828	0.339267	3.96679E-02	0.1278597	0.0318032	0.21767	300.0461	790.8252	800.7811	14.0874
46	2.86830	1.3176E-02	0.117089	0.337387	3.93466E-02	0.1257294	0.0305211	0.21329	293.1693	786.4420	796.0197	14.0690
47	2.95723	1.3882E-02	0.114257	0.335426	3.89937E-02	0.1235127	0.0292184	0.20874	286.0982	781.8727	791.0652	14.0503
48	3.04857	1.4620E-02	0.111329	0.333382	3.86072E-02	0.1212058	0.0278965	0.20401	278.7662	777.1081	785.9690	14.0315
49	3.14522	1.5392E-02	0.108302	0.331250	3.81846E-02	0.1188045	0.0265569	0.19910	271.1863	772.1386	780.5417	14.0123
50	3.23442	1.6201E-02	0.105173	0.329026	3.77235E-02	0.1163041	0.0252012	0.19400	263.3518	766.9534	774.9532	13.9925

PROFILES-- STA (33) X= 25.00000 PRESSURE= 13.6300

N	Y	PSI *	UD	TMD	** DIMENSIONLESS ** TI	TIO **	PTD	MACH **	U	** DIMENSIONAL ** T	TOI	PTOT *
51	3.34765	1.7047E-02	0.101939	0.326704	3.72214E-02	0.1136999	0.0238315	0.18870	255.2541	761.5111	769.1324	13.9728
52	3.45460	1.7933E-02	0.098598	0.324279	3.66753E-02	0.1109864	0.0224499	0.18320	246.8866	755.0889	763.0674	13.9529
53	3.56621	1.8861E-02	0.095145	0.321795	3.60821E-02	0.1081576	0.0210588	0.17798	238.2497	793.9928	756.7497	13.9323
54	3.68267	1.9831E-02	0.091577	0.319096	3.54385E-02	0.1052069	0.0196610	0.17153	229.3076	743.6074	750.1496	13.9129
55	3.80442	2.0847E-02	0.087891	0.316324	3.47408E-02	0.1021271	0.0182594	0.16535	220.0781	737.3454	743.2557	13.8925
56	3.93197	2.1910E-02	0.084083	0.313420	3.39850E-02	0.0989096	0.0168573	0.15891	210.5420	730.5772	736.0744	13.8725
57	4.06594	2.3022E-02	0.080148	0.310376	3.31664E-02	0.0955452	0.0154585	0.15222	200.6881	723.4805	728.5545	13.8524
58	4.20701	2.4187E-02	0.076080	0.307179	3.22799E-02	0.0920227	0.0140670	0.14524	190.5059	716.0297	720.6913	13.8325
59	4.35606	2.5406E-02	0.071876	0.303918	3.13198E-02	0.0883293	0.0126875	0.13797	179.9752	708.1947	712.4261	13.8125
60	4.51412	2.6682E-02	0.067527	0.300277	3.02790E-02	0.0844496	0.0113249	0.13039	169.0854	699.9399	703.7545	13.7929
61	4.68247	2.8017E-02	0.063025	0.296537	2.91499E-02	0.0803650	0.0099848	0.12246	157.8145	691.2221	694.6250	13.7736
62	4.86274	2.9415E-02	0.058362	0.292575	2.79212E-02	0.0760521	0.0086736	0.11416	146.1377	681.9875	684.9853	13.7548
63	5.05705	3.0877E-02	0.053524	0.288363	2.65818E-02	0.0714814	0.0073982	0.10546	134.0234	672.1681	674.7591	13.7364
64	5.26823	3.2408E-02	0.048494	0.283861	2.51151E-02	0.0666136	0.0061666	0.09631	121.4291	661.6744	663.8890	13.7187
65	5.50022	3.4011E-02	0.043249	0.279018	2.34989E-02	0.0613951	0.0049477	0.08663	108.2949	650.3947	652.2252	13.7017
66	5.75853	3.5688E-02	0.037752	0.273758	2.17011E-02	0.0557494	0.0039717	0.07634	94.5301	638.1254	639.6063	13.6857
67	6.05331	3.7443E-02	0.031944	0.267970	1.96702E-02	0.0495595	0.0028309	0.06529	79.9875	624.6339	625.7712	13.6707
68	6.40014	3.9281E-02	0.025718	0.261464	1.73107E-02	0.0426298	0.0018798	0.05322	64.3965	609.4687	610.2375	13.6570
69	6.83530	4.1204E-02	0.018831	0.253859	1.43784E-02	0.0345614	0.0010377	0.03954	47.1514	591.7401	592.2433	13.6449
70	7.49963	4.3216E-02	0.010212	0.243630	9.30145E-03	0.0237515	0.0003179	0.02189	25.5716	567.8974	568.0873	13.6346
71	12.13291	4.5323E-02	0.000706	0.223729	1.99865E-03	0.0029150	0.0000017	0.00158	1.7675	521.5089	521.5155	13.6300
72	15.56908	4.7528E-02	0.	0.220937	0.	-0.0000000	0.	0.	0.	515.0000	515.0000	13.6300

* JET ANALYSIS PROGRAM *

MOLE FRACTIONS-- STA (33) X= 25.00000 PRESSURE= 13.6300

N	Y	GAS1	GAS2	GAS3	GAS4	GAS5	GAS6	GAS7	GAS8	GAS9	GAS10	GAS11	GAS12
1	0.	0.003024	0.010737	0.021580	0.030954	0.023402	0.009158	0.017380	0.017449	0.006899	0.003522	0.001967	0.853928
2	0.22042	0.003024	0.010737	0.021580	0.030954	0.023402	0.009158	0.017380	0.017449	0.006899	0.003522	0.001967	0.853928
3	0.31539	0.003020	0.010721	0.021550	0.030911	0.023370	0.009146	0.017357	0.017425	0.006890	0.003517	0.001964	0.854128
4	0.39093	0.003015	0.010702	0.021511	0.030858	0.023331	0.009130	0.017327	0.017396	0.006878	0.003511	0.001961	0.854380
5	0.45697	0.003009	0.010681	0.021469	0.030798	0.023286	0.009113	0.017294	0.017363	0.006865	0.003504	0.001957	0.854662
6	0.51717	0.003002	0.010657	0.021423	0.030733	0.023237	0.009094	0.017259	0.017327	0.006851	0.003497	0.001953	0.854967
7	0.57361	0.002995	0.010632	0.021373	0.030663	0.023186	0.009074	0.017221	0.017289	0.006836	0.003489	0.001949	0.855292
8	0.62740	0.002987	0.010606	0.021321	0.030590	0.023131	0.009053	0.017180	0.017249	0.006820	0.003481	0.001945	0.855638
9	0.67925	0.002980	0.010578	0.021266	0.030512	0.023073	0.009030	0.017139	0.017206	0.006803	0.003473	0.001940	0.856002
10	0.72974	0.002971	0.010549	0.021208	0.030431	0.023012	0.009006	0.017093	0.017161	0.006785	0.003464	0.001935	0.856385
11	0.77928	0.002963	0.010518	0.021148	0.030345	0.022948	0.008981	0.017046	0.017114	0.006767	0.003454	0.001929	0.856787
12	0.82836	0.002953	0.010486	0.021084	0.030255	0.022881	0.008955	0.016997	0.017065	0.006747	0.003444	0.001924	0.857208
13	0.87657	0.002944	0.010453	0.021017	0.030161	0.022811	0.008928	0.016945	0.017013	0.006727	0.003434	0.001918	0.857649
14	0.92440	0.002934	0.010417	0.020947	0.030063	0.022738	0.008900	0.016891	0.016959	0.006705	0.003423	0.001912	0.858111
15	0.97229	0.002923	0.010381	0.020874	0.029960	0.022661	0.008870	0.016834	0.016902	0.006681	0.003411	0.001906	0.858594
16	1.02019	0.002913	0.010342	0.020798	0.029853	0.022581	0.008838	0.016775	0.016843	0.006660	0.003398	0.001899	0.859098
17	1.06820	0.002901	0.010302	0.020719	0.029741	0.022497	0.008806	0.016713	0.016781	0.006635	0.003387	0.001892	0.859625
18	1.11645	0.002889	0.010260	0.020636	0.029624	0.022410	0.008772	0.016644	0.016717	0.006610	0.003374	0.001885	0.860175
19	1.16501	0.002877	0.010216	0.020549	0.029501	0.022319	0.008736	0.016582	0.016649	0.006583	0.003360	0.001877	0.860750
20	1.21397	0.002864	0.010171	0.020459	0.029374	0.022224	0.008699	0.016512	0.016579	0.006555	0.003346	0.001869	0.861349
21	1.26342	0.002851	0.010123	0.020364	0.029241	0.022124	0.008660	0.016438	0.016506	0.006526	0.003331	0.001861	0.861974
22	1.31343	0.002837	0.010074	0.020266	0.029102	0.022021	0.008620	0.016362	0.016429	0.006496	0.003316	0.001852	0.862626
23	1.36406	0.002822	0.010022	0.020163	0.028957	0.021913	0.008578	0.016282	0.016349	0.006465	0.003300	0.001843	0.863305
24	1.41540	0.002807	0.009968	0.020057	0.028807	0.021800	0.008534	0.016199	0.016266	0.006432	0.003283	0.001834	0.864014
25	1.46750	0.002791	0.009912	0.019945	0.028649	0.021683	0.008488	0.016113	0.016179	0.006397	0.003266	0.001824	0.864752
26	1.52044	0.002774	0.009854	0.019829	0.028486	0.021560	0.008441	0.016022	0.016089	0.006362	0.003247	0.001814	0.865521
27	1.57427	0.002757	0.009793	0.019709	0.028315	0.021433	0.008391	0.015928	0.015995	0.006324	0.003228	0.001803	0.866323
28	1.62907	0.002739	0.009730	0.019583	0.028138	0.021300	0.008339	0.015830	0.015897	0.006285	0.003209	0.001792	0.867158
29	1.68490	0.002720	0.009664	0.019452	0.027952	0.021162	0.008285	0.015728	0.015794	0.006245	0.003188	0.001781	0.868028
30	1.74183	0.002701	0.009595	0.019316	0.027760	0.021018	0.008229	0.015622	0.015688	0.006203	0.003167	0.001769	0.868933
31	1.79993	0.002681	0.009524	0.019174	0.027559	0.020868	0.008171	0.015511	0.015577	0.006159	0.003144	0.001756	0.869877
32	1.85926	0.002660	0.009449	0.019026	0.027350	0.020711	0.008110	0.015396	0.015461	0.006114	0.003121	0.001743	0.870859
33	1.91990	0.002638	0.009372	0.018872	0.027133	0.020549	0.008046	0.015276	0.015341	0.006066	0.003097	0.001730	0.871881
34	1.98193	0.002615	0.009292	0.018712	0.026906	0.020379	0.007980	0.015150	0.015216	0.006017	0.003071	0.001716	0.872946
35	2.04542	0.002591	0.009208	0.018546	0.026670	0.020203	0.007911	0.015020	0.015085	0.005965	0.003045	0.001701	0.874054
36	2.11046	0.002567	0.009121	0.018372	0.026425	0.020019	0.007840	0.014885	0.014949	0.005911	0.003018	0.001686	0.875208
37	2.17713	0.002541	0.009030	0.018192	0.026170	0.019828	0.007765	0.014743	0.014808	0.005855	0.002989	0.001670	0.876409
38	2.24553	0.002514	0.008936	0.018004	0.025904	0.019629	0.007687	0.014596	0.014660	0.005797	0.002959	0.001653	0.877660
39	2.31575	0.002487	0.008838	0.017809	0.025627	0.019421	0.007607	0.014443	0.014507	0.005736	0.002928	0.001636	0.878961
40	2.38790	0.002459	0.008736	0.017606	0.025339	0.019206	0.007522	0.014283	0.014347	0.005673	0.002896	0.001618	0.880316
41	2.46208	0.002429	0.008630	0.017395	0.025039	0.018981	0.007435	0.014117	0.014180	0.005607	0.002863	0.001599	0.881727
42	2.53842	0.002396	0.008520	0.017175	0.024727	0.018747	0.007343	0.013944	0.014007	0.005539	0.002828	0.001579	0.883195
43	2.61704	0.002364	0.008405	0.016946	0.024402	0.018503	0.007248	0.013764	0.013826	0.005468	0.002791	0.001559	0.884724
44	2.69807	0.002330	0.008285	0.016708	0.024064	0.018249	0.007149	0.013576	0.013638	0.005393	0.002753	0.001538	0.886316
45	2.78167	0.002295	0.008161	0.016460	0.023712	0.017985	0.007046	0.013381	0.013442	0.005316	0.002714	0.001516	0.887973
46	2.86800	0.002259	0.008032	0.016202	0.023345	0.017709	0.006938	0.013177	0.013237	0.005235	0.002672	0.001493	0.889700
47	2.95723	0.002220	0.007897	0.015934	0.022963	0.017422	0.006826	0.012964	0.013024	0.005151	0.002629	0.001469	0.891499
48	3.04957	0.002181	0.007757	0.015654	0.022565	0.017123	0.006709	0.012743	0.012802	0.005063	0.002585	0.001444	0.893374
49	3.14522	0.002140	0.007612	0.015363	0.022150	0.016811	0.006588	0.012512	0.012571	0.004971	0.002538	0.001418	0.895328
50	3.24442	0.002097	0.007460	0.015059	0.021718	0.016486	0.006460	0.012271	0.012529	0.004876	0.002489	0.001390	0.897365

* JET ANALYSIS PROGRAM *

MOLE FRACTIONS-- STA (33) X= 25.00000 PRESSURE= 13.6300

N	Y	GAS1	GAS2	GAS3	GAS4	GAS5	GAS6	GAS7	GAS8	GAS9	GAS10	GAS11	GAS12
51	3.34785	0.002052	0.007302	0.014743	0.021266	0.016146	0.006328	0.012019	0.012077	0.004776	0.002438	0.001562	0.899491
52	3.45460	0.002006	0.007137	0.014412	0.020796	0.015792	0.006189	0.011757	0.011813	0.004672	0.002385	0.001332	0.901709
53	3.55621	0.001957	0.006965	0.014068	0.020309	0.015422	0.006044	0.011482	0.011539	0.004563	0.002330	0.001301	0.904026
54	3.68267	0.001906	0.006785	0.013709	0.019790	0.015034	0.005893	0.011195	0.011250	0.004449	0.002271	0.001269	0.906447
55	3.80442	0.001853	0.006598	0.013333	0.019253	0.014630	0.005735	0.010895	0.010949	0.004330	0.002211	0.001235	0.908979
56	3.93197	0.001798	0.006402	0.012940	0.018691	0.014205	0.005569	0.010581	0.010633	0.004205	0.002147	0.001199	0.911629
57	4.06598	0.001740	0.006197	0.012529	0.018102	0.013761	0.005395	0.010251	0.010302	0.004075	0.002080	0.001162	0.914406
58	4.20701	0.001680	0.005982	0.012097	0.017485	0.013294	0.005213	0.009904	0.009954	0.003937	0.002010	0.001123	0.917321
59	4.35606	0.001616	0.005757	0.011644	0.016835	0.012804	0.005021	0.009540	0.009588	0.003792	0.001936	0.001082	0.920355
60	4.51412	0.001550	0.005520	0.011168	0.016152	0.012286	0.004818	0.009156	0.009203	0.003640	0.001858	0.001038	0.923612
61	4.69247	0.001479	0.005270	0.010665	0.015430	0.011740	0.004604	0.008750	0.008795	0.003479	0.001776	0.000992	0.927018
62	4.86274	0.001405	0.005006	0.010134	0.014666	0.011162	0.004378	0.008320	0.008363	0.003308	0.001689	0.000943	0.930626
63	5.05705	0.001325	0.004726	0.009569	0.013853	0.010546	0.004137	0.007862	0.007904	0.003126	0.001596	0.000892	0.934453
64	5.26823	0.001242	0.004427	0.008967	0.012986	0.009888	0.003879	0.007373	0.007412	0.002932	0.001497	0.000836	0.938553
65	5.50022	0.001152	0.004106	0.008319	0.012052	0.009179	0.003601	0.006845	0.006882	0.002722	0.001390	0.000776	0.942975
66	5.75863	0.001054	0.003758	0.007616	0.011037	0.008409	0.003299	0.006272	0.006306	0.002494	0.001273	0.000711	0.947771
67	6.05331	0.000946	0.003375	0.006841	0.009918	0.007559	0.002966	0.005639	0.005669	0.002243	0.001145	0.000640	0.953050
68	6.40314	0.000825	0.002943	0.005968	0.008655	0.006597	0.002569	0.004922	0.004949	0.001958	0.001000	0.000558	0.959036
69	6.83544	0.000682	0.002432	0.004935	0.007157	0.005457	0.002142	0.004072	0.004095	0.001620	0.000827	0.000462	0.966121
70	7.49963	0.000480	0.001711	0.003473	0.005040	0.003844	0.001509	0.002869	0.002885	0.001141	0.000583	0.000325	0.976141
71	12.13291	0.000091	0.000325	0.000659	0.000957	0.000730	0.000286	0.000545	0.000548	0.000217	0.000111	0.000062	0.995471
72	15.56948	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	0.	1.000000

* JET ANALYSIS PROGRAM *

FREE JET MIXING

* SUMMARY - STATION DATA - JET PROPERTIES *

N	X	B	YJ	UC	TC	TIC	PTC	ITC	YSONIC	-J
1	0.	0.00139	1.01000	0.9230883	0.8831969	2.000000E-02	1.000000	1.000000	0.957572	1.000000
2	0.02000	0.17262	1.05210	0.9230883	0.8831974	1.998013E-02	0.931366	0.855123	751.269577	1.037279
3	0.05000	0.22383	1.05930	0.9230887	0.8831991	1.995053E-02	0.931365	0.855124	-63.09[illegible]197	1.055290
4	0.10000	0.29681	1.09323	0.9230912	0.8832055	1.990159E-02	0.931362	0.855130	-14.453575	1.111151
5	0.20000	0.90286	1.41330	0.9231069	0.8832369	1.980812E-02	0.931359	0.855165	-3.849709	1.247750
6	0.50000	0.79573	1.36577	0.9232719	0.8835365	1.955851E-02	0.931382	0.855518	-0.917425	1.293371
7	1.00000	0.98105	1.37689	0.9239196	0.8846572	1.93[illegible]E-02	0.931540	0.8[illegible]	-0.371739	1.392.[illegible]
8	1.50000	1.62277	1.94198	0.9250504	0.8865701	1.931157E-02	0.931874	0.859180	-0.227575	1.[illegible]
9	2.00000	1.48816	1.71893	0.9266640	0.8892262	1.976431E-02	0.932452	0.852423	-0.155272	1.[illegible]727
10	2.50000	1.63317	1.70911	0.9266982	0.8924775	2.077131E-02	0.933319	0.866426	-0.133172	1.[illegible]
11	2.77778	3.47771	1.73856	0.9300257	0.8945652	2.165782E-02	0.933923	0.869010	-0.123754	1.8[illegible]
12	3.50000	4.46765	2.23382	1.0170196	0.9742516	4.223765E-02	1.051323	0.951514	-1.213505	2.239456
13	4.00000	5.15355	2.57677	0.9872128	0.9374661	5.642597E-02	1.023535	0.734356	0.429375	2.6[illegible]
14	4.50000	5.29605	2.60802	0.9370549	0.8883881	7.800289E-02	0.960187	0.8[illegible]	0.359394	2.7[illegible]
15	5.00000	5.53141	2.76571	0.8700887	0.8350516	1.0[illegible]E-01	0.881749	0.794137	0.272-42	2.9[illegible]
16	5.41514	5.73972	2.86986	0.8147058	0.7956526	1.133774E-01	0.777[illegible]	0.757424		3.1[illegible]
17	5.55556	5.87646	2.93823	0.7971488	0.7837049	1.163703E-01	0.751047	0.7[illegible]40		3.1[illegible]
18	6.00000	6.10178	3.05089	0.7446123	0.7491842	1.225278E-01	0.673042	0.6[illegible]		3.3[illegible]
19	6.50000	6.39421	3.19960	0.6921204	0.7181413	1.251658E-01	0.596033	0.622532		3.53[illegible]
20	7.00000	6.69543	3.34771	0.6460194	0.6879727	1.246819E-01	0.532252	0.581532		3.7[illegible]11
21	7.50000	6.93708	3.46859	0.6055044	0.6636395	1.223633E-01	0.477684	0.546126		3.9[illegible]
22	8.00000	7.08307	3.59154	0.5698450	0.6423777	1.189550E-01	0.431549	0.515355		4.070[illegible]
23	9.00000	7.60035	3.80017	0.5095075	0.6002250	1.105947E-01	0.357962	0.463788		4.473274
24	10.00000	8.23557	4.11778	0.4600976	0.5759344	1.027072E-01	0.302206	0.421620		4.[illegible]
25	11.11111	8.95671	4.47836	0.4194573	0.5469849	9.430091E-02	0.254633	0.382383		5.[illegible]530
26	12.00000	9.45087	4.72543	0.3833208	0.5265660	8.825775E-02	0.225[illegible]	0.3[illegible]		5.7[illegible]
27	13.00000	10.11433	5.05717	0.3527285	0.5059291	8.213131E-02	0.195539	0.328563		6.1[illegible]
28	15.00000	11.19681	5.59841	0.3036128	0.4717423	7.145809E-02	0.15[illegible]	0.2[illegible]		7.[illegible]
29	17.00000	12.32363	6.16182	0.2664852	0.4447386	6.27[illegible]E-02	0.123472	0.251[illegible]		7.[illegible]
30	19.00000	13.34531	6.67265	0.2371377	0.4227350	5.567159E-02	0.102515	0.225140		8.7[illegible]
31	21.00000	14.53839	7.26920	0.2134653	0.4044320	4.986922E-02	0.086342	0.203354		9.61[illegible]
32	22.22222	15.06922	7.53461	0.2011220	0.3946804	4.683370E-02	0.078317	0.191676		10.1[illegible]752
33	25.00000	17.06156	8.53078	0.1776368	0.3757367	4.105175E-02	0.063545	0.169628		11.[illegible]77

```
EXECUTING PROGM=GMOD
TAPIN= T     TAP3T= T

NAMELIST    GMODOT
NPTS  =          11,
Y-IN  =   0.30000000E 00,
FUEL  (I)=
      1   0.31368264E-01,   0.34770985E-01,   0.38033628E-01,   0.42580351E-01,   0.41285879E-01,   0.36757839E-01,
      7   0.33271619E-01,   0.29752352E-01,   0.12151087E-01,   0.11725410E-01,   0.75437325E-02,   0.                ,
F     (I,1,1)=
      1   0.64906497E-01,   0.19390591E-01,
F     (I,2,1)=
      1   0.63919355E-01,   0.18492630E-01,
F     (I,3,1)=
      1   0.63622683E-01,   0.17794197E-01,
F     (I,4,1)=
      1   0.61953788E-01,   0.17447235E-01,
F     (I,5,1)=
      1   0.63016533E-01,   0.17423037E-01,
F     (I,6,1)=
      1   0.62575099E-01,   0.17596503E-01,
F     (I,7,1)=
      1   0.63592638E-01,   0.17743071E-01,
F     (I,8,1)=
      1   0.64985528E-01,   0.18048827E-01,
F     (I,9,1)=
      1   0.10337729E 00,   0.11455655E-01,
F     (I,10,1)=
      1   0.99527520E-01,   0.11101793E-01,
F     (I,11,1)=
      1   0.99613601E-01,   0.72312733E-02,
F     (I,12,1)=
      1   0.              ,   0.              ,
F     (I,1,2)=
      1   0.              ,   0.              ,
F     (I,2,2)=
      1   0.              ,   0.              ,
F     (I,3,2)=
      1   0.              ,   0.              ,
F     (I,4,2)=
      1   0.              ,   0.              ,
F     (I,5,2)=
      1   0.              ,   0.              ,
F     (I,6,2)=
      1   0.              ,   0.              ,
F     (I,7,2)=
      1   0.              ,   0.              ,
F     (I,8,2)=
      1   0.              ,   0.              ,
F     (I,9,2)=
      1   0.              ,   0.              ,
F     (I,10,2)=
      1   0.              ,   0.              ,
F     (I,11,2)=
      1   0.              ,   0.              ,
F     (I,12,2)=
      1   0.              ,   0.              ,
SPALOG(I)=
      1   0.40051760E-03,   0.47285956E-03,   0.51790710E-03,   0.48686348E-03,   0.48514714E-03,   0.49664912E-03,
      7   0.47094133E-03,   0.41236643E-03,   0.61355188E-04,   0.54817231E-04,   0.28768012E-04,   0.                ,
SPLOG1(I)=
      1   0.33475080E-03,   0.47285956E-03,   0.51790710E-03,   0.48686348E-03,   0.48514714E-03,   0.49664912E-03,
      7   0.47094133E-03,   0.31962430E-03,   0.11284597E-05,   0.90742828E-06,   0.22780507E-06,   0.                ,
```

```
GJE7  (I)=
    1  0.33475080E-03,  0.33475080E-03,  0.33475080E-03,  0.33475080E-03,  0.34556168E-03,  0.40380518E-03,
    7  0.45904808E-03,  0.47285956E-03,  0.47285956E-03,  0.47285956E-03,  0.47736451E-03,  0.49538353E-03,
   13  0.51340234E-03,  0.51790710E-03,  0.51790710E-03,  0.51790710E-03,  0.51480274E-03,  0.50236529E-03,
   19  0.48796745E-03,  0.48686348E-03,  0.48686348E-03,  0.48686348E-03,  0.48669155E-03,  0.48600531E-03,
   25  0.48531877E-03,  0.48514714E-03,  0.48514714E-03,  0.48514714E-03,  0.48629555E-03,  0.49097813E-03,
   31  0.49546294E-03,  0.49660912E-03,  0.49660912E-03,  0.49660912E-03,  0.49403254E-03,  0.48372523E-03,
   37  0.47341811E-03,  0.47084133E-03,  0.47084133E-03,  0.47084133E-03,  0.45571963E-03,  0.39525282E-03,
   43  0.33474630E-03,  0.31962430E-03,  0.31962430E-03,  0.31962430E-03,  0.28777471E-03,  0.16037651E-03,
   49  0.32978045E-04,  0.11284597E-05,  0.11284597E-05,  0.11284597E-05,  0.11063566E-05,  0.10179440E-05,
   55  0.92255142E-06,  0.90742828E-06,  0.90742828E-06,  0.90742828E-06,  0.83946595E-06,  0.55761667E-06,
   61  0.29575739E-06,  0.22780507E-06,  0.22780507E-06,  0.22780507E-06,  0.20502456E-06,  0.11390253E-06,
   67  0.22780507E-07,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
   73  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
   79  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
   85  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
   91  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
   97  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  103  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  109  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  115  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  121  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  127  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  133  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  139  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  145  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  151  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  157  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  163  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  169  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  175  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  181  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  187  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  193  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
  199  0.          ,  0.          ,
NJ    =         65,
RICH  (I)=
    1  0.26373036E 00,  0.35913365E 00,  0.44163468E 00,  0.56473190E 00,  0.53996278E 00,  0.42506515E 00,
    7  0.33868463E 00,  0.24935337E 00,  0.78207464E-02,  0.70844695E-02,  0.33822469E-02,  0.          ,
   13  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
   19  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,  0.          ,
$ END
```

```
EXECUTING PROGM=SPALDG
TAPIN= T     TAPOT= 7

** X=

 1   0.

PSI

 1   0.          0.1625E-03  0.3249E-03  0.4074E-03  0.6499E-03  0.8124E-03  0.9748E-03  0.001137    0.001300    0.001462
11   0.001625    0.001707    0.001950    0.002112    0.002275    0.002437    0.002600    0.002762    0.002924    0.003087
21   0.003249    0.003412    0.003574    0.003737    0.003899

R

 1   0.          0.194901    0.277517    0.340213    0.394900    0.439944    0.483001    0.521915    0.559363    0.593377
11   0.626046    0.657944    0.686931    0.715239    0.742167    0.768573    0.794168    0.818888    0.842947    0.866426
21   0.889504    0.912334    0.934978    0.964051    1.000000

G

 1   0.3348E-03  0.4729E-03  0.4729E-03  0.5179E-03  0.5179E-03  0.5179E-03  0.4985E-03  0.4869E-03  0.4869E-03  0.4569E-03
11   0.4869E-03  0.4860E-03  0.4851E-03  0.4851E-03  0.4851E-03  0.4955E-03  0.4952E-03  0.4708E-03  0.4708E-03  0.4321E-03
21   0.3196E-03  0.3196E-03  0.1545E-03  0.1000E-05  0.1139E-06

X=        0.020000

   PSI          R          G            PSI          R          G
0.          0.          0.4649E-03  0.000162    0.194935    0.4651E-03
0.000325    0.277572    0.4658E-03  0.000487    0.340273    0.5098E-03
0.000650    0.394925    0.5107E-03  0.000812    0.440026    0.5098E-03
0.000975    0.483950    0.4906E-03  0.001137    0.522007    0.4993E-03
0.001300    0.559411    0.4804E-03  0.001462    0.593405    0.4005E-03
0.001625    0.626064    0.4895E-03  0.001787    0.657946    0.4796E-03
0.001950    0.686949    0.4708E-03  0.002112    0.715261    0.4786E-03
0.002275    0.742174    0.4755E-03  0.002437    0.768575    0.4868E-03
0.002600    0.794158    0.4864E-03  0.002762    0.810891    0.4646E-03
0.002924    0.842967    0.4603E-03  0.003087    0.866450    0.4172E-03
0.003249    0.859606    0.3177E-03  0.003412    0.912593    0.2727E-03
0.003574    0.935987    0.1790E-03  0.003737    0.963303    0.7565E-04
0.003899    1.030026    0.2046E-04  0.004062    2.690500    0.4248E-05
0.004224    5.137047    0.          0.004387    0.          0.

X=        0.050200

   PSI          R          G            PSI          R          G
0.          0.          0.4485E-03  0.000162    0.194935    0.4492E-03
0.000325    0.277572    0.4516E-03  0.000407    0.340273    0.4936E-03
0.000650    0.394925    0.4964E-03  0.000812    0.440026    0.4936E-03
0.000975    0.483950    0.4739E-03  0.001137    0.522007    0.4669E-03
0.001300    0.559411    0.4677E-03  0.001462    0.593405    0.4681E-03
0.001625    0.626064    0.4680E-03  0.001787    0.657946    0.4670E-03
0.001950    0.696949    0.4663E-03  0.002112    0.715261    0.4657E-03
0.002275    0.742174    0.4648E-03  0.002437    0.768575    0.4687E-03
0.002600    0.794158    0.4659E-03  0.002752    0.810891    0.4510E-03
0.002924    0.842967    0.4387E-03  0.003087    0.846450    0.3881E-03
0.003249    0.889606    0.2976E-03  0.003412    0.912593    0.2546E-03
0.003574    0.935987    0.2043E-03  0.003737    0.963303    0.1501E-03
0.003899    1.020926    0.5113E-04  0.004062    2.690500    0.1631E-04
0.004224    5.137047    0.5678E-05  0.004387    7.523594    0.
```

ANALYTICAL MODEL OF EXHAUST PLUMES FROM AFTERBURNING ENGINES TO PREDICT REACTION OF CONTAMINANT EMISSIONS.
GENERAL ELECTRIC COMPANY — AIRCRAFT ENGINE GROUP — COMPUTED AT 23.18 HOURS ON 6/30/75 — ELAPSED TIME 1420.07 SECONDS
J85-5 MID.AB — EFTC DEMONSTRATION TEST — RUN 12-3

PROFILES AND CONTAMINANT RESIDUALS 0.068 FEET FROM NOZZLE EXIT

RADIUS, FEET INNER	RADIUS, FEET OUTER	GAS FLOW PPS	VELOCITY FPS	RICH PART FUEL/GAS	RICH PART TEMP,DEGR	LEAN PART FUEL/GAS	LEAN PART TEMP,DEGR	PREDICTED INDICATION PT,PSIA	PREDICTED INDICATION TT,DEGR
0.	1.01331E-01	1.32285E 00	2.55700E 03	6.14925E-02	3.56233E 03	1.92385E-02	1.47775E 01	2.77768E 01	2.45471E 03
1.01331E-01	2.07541E-01	4.29380E 00	2.55062E 03	6.25094E-02	3.59055E 03	1.84898E-02	1.43749E 03	2.89366E 01	2.63399E 01
2.07541E-01	3.24444E-01	8.13282E 00	2.72916E 03	6.22283E-02	3.55861E 03	1.78185E-02	1.41456E 03	2.60030E 01	2.75156E 03
3.24444E-01	4.43695E-01	1.17858E 01	2.88938E 03	6.11265E-02	3.50578E 03	1.74708E-02	1.38914E 03	2.61355E 01	2.92644E 03
4.43695E-01	5.15809E-01	8.66597E 00	2.84346E 03	6.08265E-02	3.50626E 03	1.74429E-02	1.40040E 03	2.78401E 01	2.69976E 03
5.15809E-01	5.41355E-01	3.58240E 00	2.63984E 03	6.16964E-02	3.55709E 03	1.76665E-02	1.40933E 03	2.82771E 01	2.67987E 03
5.41355E-01	5.87262E-01	6.44209E 00	2.44657E 03	6.05650E-02	3.56044E 03	1.77281E-02	1.46659E 03	2.65279E 01	2.64092E 03
5.87262E-01	6.45807E-01	8.14829E 00	1.79518E 03	5.38504E-02	3.29827E 03	1.54006E-02	1.56994E 03	2.27743E 01	2.15573E 03
6.45807E-01	7.74293E-01	1.05470E 00	1.14648E 02	3.90224E-02	2.83743E 03	1.78415E-02	1.76397E 03	1.36565E 01	2.44436E 03
7.74293E-01	2.54249E 00	1.74930E 00	2.98738E 00	1.10634E-02	1.22671E 03	8.55395E-03	1.12377E 03	1.36366E 01	1.13522E 03
2.54249E 00	3.35767E 00	1.30479E 01	1.27099E 01	3.84460E-04	5.40532E 02	2.43244E-04	5.10832E 02	1.36312E 01	5.45532E 02

AVERAGE GAS COMPOSITION UNDER FREE STREAM CONDITIONS

MASS FRACT RICH PART	FUEL/AIR RATIO	EMISSIONS INDICES, LB/KLB FUEL CO	HC	NOX AS NO2	ANALYZER READINGS, PPMV CO(DRIED)	HC AS CHN	NOX
3.09057E-01	3.30739E-02	9.12748E 01	4.12871E 01	2.53366E 00	3.19698E 03	2.71305E 03	5.07165E 01
3.73023E-01	3.62706E-02	7.05249E 01	1.97768E 01	2.66368E 00	2.72179E 03	1.42042E 03	5.83231E 01
4.59315E-01	3.98426E-02	6.33178E 01	5.91360E 00	2.97676E 00	2.63129E 03	4.64764E 02	7.13222E 01
5.64492E-01	4.40838E-02	3.36800E 01	6.62853E-01	3.11086E 00	1.50130E 03	5.74295E 01	8.23637E 01
5.47730E-01	4.30914E-02	2.91272E 01	3.92833E-01	3.08440E 00	1.34801E 03	3.33024E 01	7.97161E 01
4.65181E-01	3.69217E-02	5.36246E 01	4.41074E 00	2.61315E 00	2.11071E 03	3.22239E 02	5.82012E 01
3.80145E-01	3.53137E-02	6.21606E 01	5.91818E 00	2.59757E 00	2.33549E 03	9.35553E 02	5.54167E 01
2.52425E-01	2.53115E-02	1.01271E 02	1.87952E 01	2.72983E 00	2.66348E 03	9.41066E 02	4.36656E 01
5.52924E-01	3.05351E-02	1.09963E 02	9.58826E 00	2.49050E 00	3.25970E 03	5.82526E 02	4.61277E 01
2.94205E-01	9.45199E-03	1.40253E 02	3.15617E 01	4.13477E 00	1.37283E 03	6.07030E 02	2.42437E 01
1.00000E 00	3.85647E-04	2.10142E 02	0.	4.31241E 00	8.34070E 01	0.	9.91724E-01

AVERAGE GAS COMPOSITION AS MODIFIED BY SAMPLE PROBE

MASS FRACT RICH PART	FUEL/AIR RATIO	EMISSIONS INDICES, LB/KLB FUEL CO	HC	NOX AS NO2	ANALYZER READINGS, PPMV CO(DRIED)	HC AS CHN	NOX
2.96401E-01	3.29032E-02	9.08266E 01	4.17045E 01	2.54361E 00	3.16491E 03	2.72731E 03	5.07111E 01
3.58595E-01	3.55873E-02	6.97386E 01	2.06069E 01	2.69837E 00	2.63822E 03	1.45525E 03	5.80127E 01
4.35675E-01	3.87056E-02	6.06216E 01	6.30656E 00	3.01415E 00	2.50515E 03	4.95138E 02	7.02414E 01
5.35057E-01	4.26819E-02	3.17265E 01	7.29930E-01	3.14370E 00	1.45359E 03	6.13142E 01	8.05043E 01
5.19760E-01	4.17735E-02	2.92318E 01	4.29755E-01	3.11798E 00	1.30959E 03	3.53627E 01	7.82165E 01
3.85980E-01	3.60027E-02	5.31316E 01	4.66663E 00	2.65990E 00	2.03739E 03	3.32768E 02	5.78233E 01
3.71277E-01	3.48965E-02	6.37976E 01	6.09466E 00	2.62167E 00	2.29349E 03	4.21651E 02	5.52945E 01
2.54545E-01	2.59125E-02	1.05286E 02	1.76319E 01	2.63091E 00	2.87096E 03	9.13652E 02	4.15612E 01
6.08204E-01	3.17832E-02	1.04926E 02	8.08246E 00	2.37751E 00	3.52997E 03	5.10367E 02	4.57680E 01
2.93257E-01	9.47413E-03	1.43310E 02	3.10930E 01	4.12757E 00	1.40699E 03	5.99585E 02	2.42570E 01
1.00000E 00	3.85647E-04	2.10142E 02	0.	4.11241E 00	8.34070E 01	0.	9.93724E-01

INTEGRATED PROFILE AVERAGES (FREE STREAM CONDITIONS)

TOTAL FLOWS, PPS GAS MIX	FUEL	EMISSIONS INDICES, LB/KLB FUEL CO	HC	NOX AS NO2	CONTAMINANT FLOWS, PPS CO	HC	NOX AS NO2
6.84258E 01	1.96183E 00	5.44751E 01	6.84221E 00	2.91336E 00	1.06873E-01	1.34232E-02	5.71318E-03

ANALYTICAL MODEL OF EXHAUST PLUMES FROM AFTERBURNING ENGINES TO PREDICT REACTION OF CONTAMINANT EMISSIONS.

GENERAL ELECTRIC COMPANY   AIRCRAFT ENGINE GROUP   COMPUTED AT 23.19 HOURS ON 6/30/75   ELAPSED TIME 1460.67 SECONDS

J85-5 MID.AB   EFTC DEMONSTRATION TEST   RUN 32-3

PROFILES AND CONTAMINANT RESIDUALS 0.675 FEET FROM NOZZLE EXIT

RADIUS, FEET		GAS FLOW	VELOCITY	RICH PART		LEAN PART		PREDICTED INDICATION	
INNER	OUTER	PPS	FPS	FUEL/GAS	TEMP,DEGR	FUEL/GAS	TEMP,DEGR	PT,PSIA	TT,DEGR
0.	9.98457E-02	1.28325E 00	2.45923E 03	4.89388E-02	3.03471E 03	1.90440E-02	1.47246E 03	2.68301E 01	2.42345E 03
9.98457E-02	2.06302E-01	4.24100E 00	2.60831E 03	4.94992E-02	3.03256E 03	1.84567E-02	1.43774E 03	2.73444E 01	2.52668E 03
2.06302E-01	3.23321E-01	8.08585E 00	2.77305E 03	5.17508E-02	3.11282E 03	1.18652E-02	1.41208E 03	2.61582E 01	2.67160E 03
3.23321E-01	4.43850E-01	1.19869E 01	2.90191E 03	5.35835E-02	3.18185E 03	1.75237E-02	1.38749E 03	2.88530E 01	2.80085E 03
4.43850E-01	5.20511E-01	9.60672E 00	2.74228E 03	5.09327E-02	3.08586E 03	1.71907E-02	1.36741E 03	2.78007E 01	2.65031E 03
5.20511E-01	5.51247E-01	4.18425E 00	2.35521E 03	4.31667E-02	2.79596E 03	1.63080E-02	1.40373E 03	2.56546E 01	2.32152E 03
5.51247E-01	6.25416E-01	1.00870E 01	1.78632E 03	2.56492E-02	1.96753E 03	1.58590E-02	1.50374E 03	2.15922E 01	1.83762E 03
6.25416E-01	8.22110E-01	1.56583E 01	6.40069E 02	1.18884E-02	1.26203E 03	1.57096E-02	1.63743E 03	1.44267E 01	1.35063E 03
8.22110E-01	2.22627E 00	7.37852E 00	1.22540E 01	3.94414E-03	7.98390E 02	1.56740E-02	1.69603E 03	1.36307E 01	8.47341E 02
2.22627E 00	3.60538E 00	1.21859E 00	1.92169E 00	1.33458E-02	1.45533E 03	1.56740E-02	1.69986E 03	1.36300E 01	1.44276E 03
3.60538E 00	4.18464E 00	1.06953E 01	1.12858E 01	4.69022E-04	5.49354E 02	0.	5.14605E 02	1.36309E 01	5.49354E 02

AVERAGE GAS COMPOSITION UNDER FREE STREAM CONDITIONS

MASS FRACT	FUEL/AIR	EMISSIONS INDICES, LB/KLB FUEL			ANALYZER READINGS, PPMV		
RICH PART	RATIO	CO	HC	NOX AS NO2	CO(DRIED)	HC AS CHN	NOX
4.68044E-01	3.41288E-02	6.56705E 01	2.71141E 01	2.58599E 00	2.37636E 03	1.93622E 03	5.33894E 01
5.44332E-01	3.61338E-02	5.69766E 01	1.37711E 01	2.72164E 00	2.22823E 03	1.00112E 03	6.03185E 01
6.37147E-01	4.00833E-02	4.69000E 01	4.76033E 00	2.96864E 00	2.00965E 03	3.76301E 02	7.15413E 01
6.62320E-01	4.33123E-02	3.02137E 01	1.01346E 00	3.08378E 00	1.40520E 03	8.63244E 01	8.00776E 01
6.17528E-01	3.96373E-02	3.34109E 01	1.81841E 00	2.95124E 00	1.41636E 03	1.42260E 02	7.03876E 01
5.24900E-01	3.14444E-02	4.35084E 01	5.29819E 00	2.85671E 00	1.45052E 03	3.31534E 02	5.44964E 01
5.99400E-01	2.22698E-02	6.91424E 01	7.73445E 00	2.77790E 00	1.61780E 03	3.46003E 02	3.76850E 01
7.88019E-01	1.25616E-02	8.75855E 01	7.44593E 00	2.75266E 00	1.17168E 03	1.94205E 02	2.15874E 01
9.76110E-01	4.25374E-03	1.03678E 02	2.55818E 00	2.75200E 00	4.55010E 02	2.22712E 01	7.30399E 00
9.74364E-01	1.36243E-02	8.71394E 01	8.26078E 00	2.74309E 00	1.23587E 03	2.28053E 02	2.30863E 01
1.00000E 00	4.70509E-04	1.18924E 02	4.91667E 00	2.91109E 00	5.75853E 01	4.75437E 00	8.58179E-01

AVERAGE GAS COMPOSITION AS MODIFIED BY SAMPLE PROBE

MASS FRACT	FUEL/AIR	EMISSIONS INDICES, LB/KLB FUEL			ANALYZER READINGS, PPMV		
RICH PART	RATIO	CO	HC	NOX AS NO2	CO(DRIED)	HC AS CHN	NOX
4.51268E-01	3.37195E-02	6.50813E 01	2.80863E 01	2.61196E 00	2.32694E 03	1.98012E 03	5.33038E 01
5.24929E-01	3.61002E-02	5.65097E 01	1.45928E 01	2.75170E 00	2.17021E 03	1.04315E 03	5.99665E 01
5.83255E-01	3.92050E-02	4.66406E 01	5.15848E 00	2.99688E 00	1.95310E 03	3.99219E 02	7.07064E 01
6.36097E-01	4.22812E-02	3.02728E 01	1.11171E 00	3.10922E 00	1.37298E 03	9.30352E 01	7.88989E 01
5.93417E-01	3.87556E-02	3.34133E 01	1.97530E 00	2.98011E 00	1.38374E 03	1.51237E 02	6.95595E 01
5.13860E-01	3.11282E-02	4.33910E 01	5.47469E 00	2.87548E 00	1.43161E 03	3.39248E 02	5.43211E 01
6.04345E-01	2.23205E-02	6.93596E 01	7.62198E 00	2.76768E 00	1.62666E 03	3.41729E 02	3.76295E 01
7.70465E-01	1.29337E-02	8.60338E 01	8.00087E 00	2.80140E 00	1.15787E 03	2.10987E 02	2.23997E 01
9.66065E-01	4.37292E-03	1.01049E 02	3.53536E 00	2.81572E 00	4.55952E 02	3.16570E 01	7.65161E 00
9.72062E-01	1.16299E-02	8.70257E 01	8.30776E 00	2.74702E 00	1.23454E 03	2.29426E 02	2.31271E 01
1.00000E 00	4.70509E-04	1.18924E 02	4.91667E 00	2.91109E 00	5.75853E 01	4.75437E 00	8.59179E-01

INTEGRATED PROFILE AVERAGES (FREE STREAM CONDITIONS)

TOTAL FLOWS, PPS		EMISSIONS INDICES, LB/KLB FUEL			CONTAMINANT FLOWS, PPS		
GAS MIX	FUEL	CO	HC	NOX AS NO2	CO	HC	NOX AS NO2
8.44259E 01	1.96183E 00	4.91388E 01	5.06798E 00	2.91155E 00	9.64017E-02	9.94152E-03	5.71196E-03

ANALYTICAL MODEL OF EXHAUST PLUMES FROM AFTERBURNING ENGINES TO PREDICT REACTION OF CONTAMINANT EMISSIONS,
GENERAL ELECTRIC COMPANY AIRCRAFT ENGINE GROUP COMPUTED AT 23.23 HOURS ON 6/30/75 ELAPSED TIME 1595.[illegible] SECONDS
J85-5 MID.AB EPTC DEMONSTRATION TEST RUN 32.[illegible]

PROFILES AND CONTAMINANT RESIDUALS 3.750 FEET FROM NOZZLE EXIT

RADIUS, FEET INNER	RADIUS, FEET OUTER	GAS FLOW PPS	VELOCITY FPS	RICH PART FUEL/GAS	RICH PART TEMP.DEGR	LEAN PART FUEL/GAS	LEAN PART TEMP.DEGR	PREDICTED INDICATION PT,PSIA	PREDICTED INDICATION TT,DEGR
0.	9.73499E-02	1.21344E 00	2.65918E 03	3.52936E-02	2.41956E 03	1.86239E-02	1.45689E 03	2.95260E 01	2.43951E 03
9.73494E-02	2.03559E-01	4.16425E 00	2.74874E 03	3.64991E-02	2.45751E 03	1.82164E-02	1.43186E 03	3.05800E 01	2.43468E 03
2.03554E-01	3.27797E-01	8.65776E 00	2.74028E 03	3.66719E-02	2.44889E 03	1.72992E-02	1.39025E 03	3.05315E 01	2.42113E 03
3.27797E-01	4.98333E-01	1.75308E 01	2.25952E 03	3.00092E-02	2.17949E 03	1.66869E-02	1.44063E 03	2.57032E 01	2.11297E 03
4.98333E-01	6.40942E-01	1.75247E 01	1.67572E 03	2.16421E-02	1.81783E 03	1.65175E-02	1.64278E 03	2.08922E 01	1.79272E 03
6.40942E-01	7.00793E-01	7.60363E 00	1.31328E 03	1.67728E-02	1.58166E 03	1.64665E-02	1.71101E 03	1.84790E 01	1.59087E 03
7.00783E-01	8.42291E-01	1.64140E 01	9.22175E 02	1.29960E-02	1.37568E 03	1.64141E-02	1.74741E 03	1.62357E 01	1.[illegible]910E 03
8.42291E-01	1.37614E 00	2.56566E 01	2.00299E 02	7.35887E-03	1.03546E 03	1.60359E-02	1.75719E 03	1.37036E 01	1.03962E 03
1.37614E 00	1.62848E 00	8.05738E 00	7.38194E 01	3.79677E-03	7.94659E 02	6.95591E-03	1.08568E 03	1.36572E 01	6.03450E 02
1.62849E 00	1.82249E 00	7.10734E 00	6.31717E 01	2.37376E-03	6.91806E 02	7.56555E-04	5.75308E 02	1.36529E 01	6.87436E 02
1.82249E 00	1.94024E 00	4.65778E 00	5.42600E 01	1.20398E-03	6.02347E 02	1.91390E-05	5.16153E 02	1.36493E 01	5.97235E 02

AVERAGE GAS COMPOSITION UNDER FREE STREAM CONDITIONS

MASS FRACT RICH PART	FUEL/AIR RATIO	EMISSIONS INDICES, LB/KLB FUEL CO	HC	NOX AS NO2	ANALYZER READINGS, PPMV CO(DRIED)	HC AS CHN	NOX
9.71010E-01	3.61632E-02	3.42913E 01	1.76241E 00	2.71242E 00	1.32221E 03	1.26252E 02	5.92167E 01
9.72291E-01	3.74411E-02	2.89927E 01	1.04272E 00	2.80494E 00	1.15914E 03	7.72442E 01	6.33468E 01
9.60135E-01	3.73369E-02	2.32683E 01	6.81475E-01	2.94251E 00	9.27781E 02	5.03921E 01	6.62847E 01
8.72659E-01	2.92164E-02	2.72001E 01	1.59201E 00	2.96634E 00	8.41576E 02	9.28115E 01	5.27203E 01
8.44638E-01	2.13472E-02	3.53879E 01	2.89034E 00	2.93248E 00	7.93456E 02	1.24101E 02	3.83850E 01
8.68343E-01	1.70631E-02	4.04320E 01	3.45968E 00	2.91271E 00	7.21409E 02	1.19254E 02	3.06079E 01
8.95807E-01	1.35644E-02	4.59776E 01	3.98854E 00	2.89274E 00	6.49974E 02	1.09722E 02	2.42599E 01
9.57317E-01	7.81048E-03	5.64695E 01	4.92490E 00	2.85373E 00	4.56795E 02	7.84465E 01	1.39577E 01
9.77294E-01	3.89396E-03	6.21721E 01	5.38154E 00	2.83400E 00	2.44764E 02	4.29128E 01	6.88932E 00
9.53135E-01	2.30447E-03	6.56965E 01	5.64902E 00	2.82230E 00	1.56296E 02	2.57665E 01	4.07681E 00
8.92811E-01	1.08105E-03	6.79203E 01	5.81281E 00	2.81508E 00	7.55436E 01	1.29067E 01	1.90554E 00

AVERAGE GAS COMPOSITION AS MODIFIED BY SAMPLE PROBE

MASS FRACT RICH PART	FUEL/AIR RATIO	EMISSIONS INDICES, LB/KLB FUEL CO	HC	NOX AS NO2	ANALYZER READINGS, PPMV CO(DRIED)	HC AS CHN	NOX
9.69049E-01	3.61280E-02	3.42884E 01	1.83307E 00	2.71413E 00	1.32076E 03	1.31192E 02	5.92185E 01
9.70354E-01	3.73440E-02	2.89963E 01	1.09666E 00	2.80669E 00	1.15792E 03	8.11522E 01	6.33176E 01
9.56414E-01	3.72696E-02	2.32841E 01	7.20112E-01	2.94502E 00	9.26674E 02	5.31179E 01	6.62761E 01
8.69657E-01	2.91739E-02	2.72125E 01	1.67531E 00	2.96949E 00	8.40697E 02	9.44189E 01	5.27018E 01
8.46219E-01	2.13557E-02	3.53911E 01	2.86651E 00	2.93014E 00	7.93849E 02	1.23125E 02	3.83692E 01
8.64182E-01	1.70624E-02	4.04144E 01	3.49869E 00	2.91663E 00	7.21065E 02	1.20594E 02	3.06079E 01
8.87519E-01	1.35975E-02	4.58382E 01	4.16144E 00	2.91042E 00	6.49363E 02	1.14712E 02	2.44581E 01
9.45931E-01	7.91109E-03	5.54026E 01	5.30837E 00	2.89396E 00	4.58080E 02	8.56353E 01	1.42326E 01
9.73694E-01	3.90544E-03	6.19808E 01	5.48401E 00	2.84502E 00	2.49733E 02	4.38580E 01	6.93640E 00
9.56930E-01	2.31610E-03	6.57295E 01	5.62863E 00	2.82007E 00	1.56815E 02	2.67406E 01	4.08440E 00
9.00140E-01	1.08478E-03	6.79254E 01	5.81059E 00	2.81485E 00	7.61597E 01	1.30058E 01	1.92074E 00

INTEGRATED PROFILE AVERAGES (FREE STREAM CONDITIONS)

TOTAL FLOWS, PPS GAS MIX	FUEL	EMISSIONS INDICES, LB/KLB FUEL CO	HC	NOX AS NO2	CONTAMINANT FLOWS, PPS CO	HC	NOX AS NO2
1.18588E 02	1.96183E 00	3.52862E 01	2.48183E 00	2.91169E 00	6.92254E-02	4.86891E-03	5.71223E-03

ANALYTICAL MODEL OF EXHAUST PLUMES FROM AFTERBURNING ENGINES TO PREDICT REACTION OF CONTAMINANT EMISSIONS.

GENERAL ELECTRIC COMPANY AIRCRAFT ENGINE GROUP COMPUTED AT 23.27 HOURS ON 6/30/75 ELAPSED TIME 1725.92 SECONDS

J85-5 MI2.AB EFTC DEMONSTRATION TEST RUN 52-3

PROFILES AND CONTAMINANT RESIDUALS 7.500 FEET FROM NOZZLE EXIT

RADIUS, FEET INNER	RADIUS, FEET OUTER	GAS FLOW PPS	VELOCITY FPS	RICH PART FUEL/GAS	RICH PART TEMP.DEGR	LEAN PART FUEL/GAS	LEAN PART TEMP.DEGR	PREDICTED INDICATION PT.PSIA	PREDICTED INDICATION TT.DEGR
0.	1.18391E-01	1.79137E 00	2.06968E 03	2.91191E-02	1.89071E 03	1.61971E-02	1.51162E 03	2.51244E 01	1.85932E 03
1.18391E-01	2.59981E-01	6.63590E 00	1.97408E 03	2.30685E-02	1.85024E 03	1.61736E-02	1.53741E 03	2.41295E 01	1.85146E 03
2.59481E-01	4.49882E-01	1.55369E 01	1.72196E 03	2.02946E-02	1.73511E 03	1.61121E-02	1.59743E 03	2.17503E 01	1.72628E 03
9.49982E-01	7.17161E-01	3.08729E 01	1.32828E 03	1.60824E-02	1.54003E 03	1.59921E-02	1.66821E 03	1.87523E 01	1.54755E 03
7.17161E-01	9.41979E-01	2.99383E 01	9.35591E 02	1.20549E-02	1.32907E 03	1.58505E-02	1.71592E 03	1.64275E 01	1.34795E 03
9.41979E-01	1.03632E 00	1.27067E 01	7.18593E 02	9.86291E-03	1.20377E 03	1.58045E-02	1.73677E 03	1.54456E 01	1.22352E 03
1.03632E 00	1.24617E 00	2.67543E 01	5.39132E 02	8.08138E-03	1.07550E 03	1.57509E-02	1.74751E 03	1.45924E 01	1.10911E 03
1.29617E 00	1.61155E 00	2.72776E 01	2.38061E 02	7.18670E-03	1.04408E 03	1.57357E-02	1.75898E 03	1.39963E 01	1.05573E 03
1.61165E 00	1.77495E 00	8.96274E 00	1.09242E 02	3.97723E-03	7.77571E 02	1.57305E-02	1.76298E 03	1.36999E 01	7.77664E 02
1.77495E 00	1.99718E 00	2.01407E 00	3.04560E 01	8.11048E-03	1.11799E 03	1.57305E-02	1.76381E 03	1.36333E 01	1.11799E 03
1.94718E 00	2.01718E 00	1.72166E 01	2.89279E 02	2.91029E-04	5.37042E 02	0.	5.14813E 02	1.42577E 01	5.37042E 02

AVERAGE GAS COMPOSITION UNDER FREE STREAM CONDITIONS

MASS FRACT RICH PART	FUEL/AIR RATIO	EMISSIONS INDICES, LB/KLB FUEL CO	HC	NOX AS NO2	ANALYZER READINGS, PPMV CO(DRIED)	HC AS CHN	NOX
9.31919E-01	2.92148E-02	2.58299E 01	1.33419E 00	2.90828E 00	6.59012E 02	6.97962E 01	4.30592E 01
9.30482E-01	2.31736E-02	2.68348E 01	1.95971E 00	2.91028E 00	6.94753E 02	6.79164E 01	9.12803E 01
9.30662E-01	2.04681E-02	2.81105E 01	1.77575E 00	2.91426E 00	6.03865E 02	7.31752E 01	3.66110E 01
9.43371E-01	1.63841E-02	3.09421E 01	2.16225E 00	2.91453E 00	5.29812E 02	7.16208E 01	2.94307E 01
9.62213E-01	1.23823E-02	3.38574E 01	2.97981E 00	2.91155E 00	4.36313E 02	6.23517E 01	2.23108E 01
9.73797E-01	1.01412E-02	3.49139E 01	2.60389E 00	2.91012E 00	3.67652E 02	5.37287E 01	1.83051E 01
9.85639E-01	8.29155E-03	3.58545E 01	2.71306E 00	2.90882E 00	3.07749E 02	4.58041E 01	1.49714E 01
9.90899E-01	7.33750E-03	3.59531E 01	2.72912E 00	2.90859E 00	2.73373E 02	4.08526E 01	1.32766E 01
9.99963E-01	3.49923E-03	3.67265E 01	2.81525E 00	2.90725E 00	1.32551E 02	2.01529E 01	6.55023E 00
1.00000E 00	8.19928E-03	3.62969E 01	2.76412E 00	2.90822E 00	3.08408E 02	4.62061E 01	1.48207E 01
1.00000E 00	2.91900E-04	4.27198E 01	3.45993E 00	2.89313E 00	1.28361E 01	2.07611E 00	5.29238E-01

AVERAGE GAS COMPOSITION AS MODIFIED BY SAMPLE PROBE

MASS FRACT RICH PART	FUEL/AIR RATIO	EMISSIONS INDICES, LB/KLB FUEL CO	HC	NOX AS NO2	ANALYZER READINGS, PPMV CO(DRIED)	HC AS CHN	NOX
9.31391E-01	2.42100E-02	2.58328E 01	1.34115E 00	2.90890E 00	6.58963E 02	6.51226E 01	4.30608E 01
9.30931E-01	2.31733E-02	2.64351E 01	1.46039E 00	2.91034E 00	6.94749E 02	6.79470E 01	9.12806E 01
9.31142E-01	2.09702E-02	2.81083E 01	1.76851E 00	2.91360E 00	6.03882E 02	7.28842E 01	3.65063E 01
9.42384E-01	1.63840E-02	3.09445E 01	2.17988E 00	2.91618E 00	5.29850E 02	7.22042E 01	2.94472E 01
9.58930E-01	1.23951E-02	3.38598E 01	2.55385E 00	2.91869E 00	4.36735E 02	6.42582E 01	2.23879E 01
9.71031E-01	1.01641E-02	3.99039E 01	2.70487E 00	2.91987E 00	3.68383E 02	5.59372E 01	1.84085E 01
9.82513E-01	8.30609E-03	3.58387E 01	2.81329E 00	2.91856E 00	3.08520E 02	4.76354E 01	1.50655E 01
9.89419E-01	7.35929E-03	3.59683E 01	2.81823E 00	2.91726E 00	2.74070E 02	4.23200E 01	1.33550E 01
9.99945E-01	3.49946E-03	3.67265E 01	2.81658E 00	2.90738E 00	1.32558E 02	2.01937E 01	6.35472E 00
1.00000E 00	8.19928E-03	3.62964E 01	2.76412E 00	2.90822E 00	3.08408E 02	4.62061E 01	1.48207E 01
1.00000E 00	2.91900E-09	4.27199E 01	3.45993E 00	2.89313E 00	1.28361E 01	2.07611E 00	5.29238E-01

INTEGRATED PROFILE AVERAGES (FREE STREAM CONDITIONS)

TOTAL FLOWS, PPS GAS MIX	FUEL	EMISSIONS INDICES, LB/KLB FUEL CO	HC	NOX AS NO2	CONTAMINANT FLOWS, PPS CO	HC	NOX AS NO2
1.79757E 02	1.96183E 00	3.20694E 01	2.25482E 00	2.91172E 00	6.29098E-02	9.42357E-03	5.71230E-03

ANALYTICAL MODEL OF EXHAUST PLUMES FROM AFTERBURNING ENGINES TO PREDICT REACTION OF CONTAMINANT EMISSIONS.

GENERAL ELECTRIC COMPANY AIRCRAFT ENGINE GROUP COMPUTED AT 23.31 HOURS ON 6/30/75 ELAPSED TIME 1855.76 SECONDS

J85-5 MOD.A9 EFTC DEMONSTRATION TEST RUN 32-3

PROFILES AND CONTAMINANT RESIDUALS 15.000 FEET FROM NOZZLE EXIT

RADIUS, FEET INNER	RADIUS, FEET OUTER	GAS FLOW PPS	VELOCITY FPS	RICH PART FUEL/GAS	RICH PART TEMP.DEGR	LEAN PART FUEL/GAS	LEAN PART TEMP.DEGR	PREDICTED INDICATION PT.PSIA	PREDICTED INDICATION TT.DEGR
0.	1.94850E-01	3.57720E 00	1.06030E 03	1.17037E-02	1.28589E 03	1.43907E-02	1.59943E 03	1.74295E 01	1.30043E 03
1.94850E-01	4.24499E-01	1.30549E 01	1.01979E 03	1.13743E-02	1.26936E 03	1.42135E-02	1.58226E 03	1.71694E 01	1.28414E 03
4.24499E-01	7.23948E-01	2.92902E 01	9.12477E 02	1.04972E-02	1.22359E 03	1.37052E-02	1.55896E 03	1.65266E 01	1.23952E 03
7.23949E-01	1.12983E 00	5.48909E 01	7.25403E 02	8.92560E-03	1.13533E 03	1.26589E-02	1.50254E 03	1.55576E 01	1.15066E 03
1.12983E 00	1.46049E 00	5.03619E 01	5.28549E 02	7.14446E-03	1.02660E 03	1.14352E-02	1.42673E 03	1.47392E 01	1.04111E 03
1.46049E 00	1.59805E 00	2.08722E 01	4.13215E 02	5.99684E-03	9.51853E 02	1.07395E-02	1.38463E 03	1.43591E 01	9.652[illegible]E 02
1.59805E 00	1.90137E 00	4.30673E 01	3.15754E 02	5.00623E-03	8.84448E 02	1.01765E-02	1.34271E 03	1.40761E 01	8.[illegible]E 02
1.90137E 00	2.43352E 00	5.77504E 01	1.68822E 02	3.41893E-03	7.72234E 02	9.36995E-03	1.28511E 03	1.37766E 01	7.74522E 02
2.43382E 00	2.66782E 00	1.50759E 01	7.27162E 01	2.06752E-03	6.72422E 02	9.55273E-03	1.28576E 03	1.36612E 01	6.72422E 02
2.66782E 00	2.95399E 00	1.17759E 01	3.93997E 01	1.38723E-03	6.21018E 02	9.68305E-11	5.14823E 02	1.36349E 01	6.21018E 02
2.95399E 00	3.32251E 00	5.40676E 00	1.18735E 01	9.27790E-04	5.86047E 02	1.44409E-10	5.14949E 02	1.36310E 01	5.86047E 02

AVERAGE GAS COMPOSITION UNDER FREE STREAM CONDITIONS

MASS FRACT RICH PART	FUEL/AIR RATIO	EMISSIONS INDICES, LB/KLB FUEL CO	HC	NOX AS NO2	ANALYZER READINGS, PPMV CO(DRIED)	HC AS CHN	NOX
9.61120E-01	1.19815E-02	3.07662E 01	2.13509E 00	2.91221E 00	3.83507E 02	5.19525E 01	2.16030E 01
9.62317E-01	1.16468E-02	3.08602E 01	2.14536E 00	2.91217E 00	3.73804E 02	5.07618E 01	2.10065E 01
9.64411E-01	1.07542E-02	3.11152E 01	2.17330E 00	2.91207E 00	3.47688E 02	4.75254E 01	1.94137E 01
9.68804E-01	9.14952E-03	3.15753E 01	2.22401E 00	2.91188E 00	2.99690E 02	4.14461E 01	1.65432E 01
9.74497E-01	7.32661E-03	3.20364E 01	2.27537E 00	2.91167E 00	2.43034E 02	3.40191E 01	1.32713E 01
9.79365E-01	6.14949E-03	3.22957E 01	2.30446E 00	2.91154E 00	2.05392E 02	2.89538E 01	1.11522E 01
9.84475E-01	5.12629E-03	3.25463E 01	2.33253E 00	2.91141E 00	1.72357E 02	2.44573E 01	9.30607E 00
9.97493E-01	3.45498E-03	3.29253E 01	2.37542E 00	2.91120E 00	1.17325E 02	1.68152E 01	6.26251E 00
1.00000E 00	2.07739E-03	3.30715E 01	2.39197E 00	2.91112E 00	7.07590E 01	1.01956E 01	3.78284E 00
1.00000E 00	1.39291E-03	3.31441E 01	2.40020E 00	2.91107E 00	4.75158E 01	6.86468E 00	2.53820E 00
1.00000E 00	9.31159E-04	3.31595E 01	2.40307E 00	2.91106E 00	3.17738E 01	4.59675E 00	1.69760E 00

AVERAGE GAS COMPOSITION AS MODIFIED BY SAMPLE PROBE

MASS FRACT RICH PART	FUEL/AIR RATIO	EMISSIONS INDICES, LB/KLB FUEL CO	HC	NOX AS NO2	ANALYZER READINGS, PPMV CO(DRIED)	HC AS CHN	NOX
9.58719E-01	1.19881E-02	3.07738E 01	2.18698E 00	2.91707E 00	3.83815E 02	5.32403E 01	2.16509E 01
9.59475E-01	1.16542E-02	3.08680E 01	2.20109E 00	2.91740E 00	3.74139E 02	5.21131E 01	2.10575E 01
9.61516E-01	1.07637E-02	3.11235E 01	2.23921E 00	2.91827E 00	3.48091E 02	4.90095E 01	1.94720E 01
9.65497E-01	9.16250E-03	3.15935E 01	2.30721E 00	2.91973E 00	3.00197E 02	4.30569E 01	1.66111E 01
9.70501E-01	7.34275E-03	3.20436E 01	2.37627E 00	2.92121E 00	2.43628E 02	3.56054E 01	1.33439E 01
9.75725E-01	6.16717E-03	3.23020E 01	2.41512E 00	2.92203E 00	2.06026E 02	3.04309E 01	1.12243E 01
9.81202E-01	5.14314E-03	3.25509E 01	2.44244E 00	2.92183E 00	1.72960E 02	2.56924E 01	9.36992E 00
9.96877E-01	3.45910E-03	3.29260E 01	2.40692E 00	2.91420E 00	1.17467E 02	1.70584E 01	6.29645E 00
1.00000E 00	2.07739E-03	3.30715E 01	2.39197E 00	2.91112E 00	7.07590E 01	1.01956E 01	3.78284E 00
1.00000E 00	1.39291E-03	3.31441E 01	2.40020E 00	2.91107E 00	4.75158E 01	6.86468E 00	2.53820E 00
1.00000E 00	9.31159E-04	3.31695E 01	2.40307E 00	2.91106E 00	3.17738E 01	4.59675E 00	1.69760E 00

INTEGRATED PROFILE AVERAGES (FREE STREAM CONDITIONS)

TOTAL FLOWS, PPS GAS MIX	FUEL	EMISSIONS INDICES, LB/KLB FUEL CO	HC	NOX AS NO2	CONTAMINANT FLOWS, PPS CO	HC	NOX AS NO2
3.05143E 02	1.96183E 00	3.18488E 01	2.25487E 00	2.91174E 00	6.24818E-02	4.42366E-03	5.71232E-03

3.4 Helpful Hints

At values of x/dj specified by variable X in Namelist $A for JETINP, JETMIX not only prints its mixing profiles, but also writes profile data on a file for tranmission to links SPALDG and EMIS. Suggested values for X are: 0.0, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0, 5.5, 6.0, 6.5, 7.0, 7.5, 8.0, 9.0, 10.0, 11.0, 12.0, 13.0, 15.0, 17.0, 19.0, 21.0, 23.0, and 25.0. Jet diameter (dj) is twice the value input for RADJ or AMBT(5) in Namelist $TSTDAT. Emissions print stations need not correspond to JETMIX print stations. Emissions reactions should be frozen by x/dj = 25.0, but several intermediate stations should be printed to verify this.

The recommended value for CXPC in Namelist $A for JETINP is 0.15. This value is chosen as a compromise between computational accuracy and reasonable computer time. Occasionally, a malfunction may result in JETMIX due to addition of excessively large streamtubes at the outer boundary, prior to reaching the end of the potential core. If this occurs, try CXPC = 0.015 and restart JETMIX from the last print station having reasonable streamtube size ($\Delta\psi$) at the outer boundary.

To restart JETMIX, delete the input for PREJET (Namelist $TSTDAT and $A for JETPRF and the preceding formatted cards), and set the variable RESTRT in $A for JETINP equal to the value of X at which restart is to occur.

The PLUMOD version of GCKP has more ability to recover from computational malfunctions than the standard NASA edition; even so, problems may arise. The initial conditions for each kinetics calculation are recorded on scratch file 40, and should be dumped automatically in the event of a failure in GCKP. These data are sufficient to permit running the troublesome calculation on a stand-along version of GCKP, if available. It is usually possible to circumvent the problem by adjustment of the step size controls in Namelist $PROB.

SECTION 4.0

SOURCE LISTINGS

Listings of the FORTRAN source decks of the various subprograms follow, in alphabetical order by deck label. To find a deck containing a given entry name (SYMDEF), the user may refer to Table 1. The locations of the various decks in the program overlay structure is given in Sections 2.1 and 2.3 above.

```
CADJMC          MW OF RAW FUEL// ADJUST C2H4 THERMO COEF
      SUBROUTINE ADJMC
      COMMON /INDATA/ N,HF,WAR,T2,BETA,T25,FAR5,EINO2C,P0,
     *                RCO(11),RCO2(11),RHC(11),RNOX(11),
     *                PT(11),PS(11),BLOC(11),QCO(11)
      REAL            N
      COMMON /TCOF  / TC(7,2,25),TLOW,TMID,THI
      COMMON /SNMW  / ALSP(2,75), ALMW(75)
C
C     CALCULATE MW OF FUEL
C
      ALMW(16)= 10.*(12.01+1.008*N)
C
C     ADJUST C2H4 THERMO COEFFICIENT
C
    2 HV     = (184686.04+37977.7*N)/(11.91968+N)
      ADD    = (HV-20275.+155.)*28.054/1.98596
      TC(6,1,16)= TC(6,1,16)+ADD
      TC(6,2,16)= TC(6,2,16)+ADD
C
    3 RETURN
      END
```

```
CALFA2      SOLVE FOR MASS TRANSFER MATRIX
C
      SUBROUTINE ALFA2 (N)
C
      COMMON/CMXFLT/W(12,12),WHAT(13,12),ALFA(12,12)
      COMMON/CNWNET/DUM1(100),WFCUM(12),DUM2(100)
      COMMON/CMASS /WT(12),WTHAT(12),DUM3(24)
C
      DIMENSION WF(11)
C
      NAMELIST /BADALF/ W,ALFA,WHAT
C
C   N IS NUMBER OF FUEL-BEARING TUBES
C   W(K,J) IS FLOW OF SPECIES K IN TUBE J AT OLD STATION
C   WHAT(K,I) IS FLOW OF SPECIES K IN TUBE I AT NEW STATION
C   ALFA(J,I) IS FRACTION OF FLOW IN TUBE J AT OLD STATION
C      THAT IS TRANSFERRED TO TUBE I AT NEW STATION
C   WF(J) IS FUEL FLOW IN TUBE J (CONSTANT BY TUBE DEFINITION)
C   WFCUM IS CUMULATIVE FUEL FLOW
C
      IF(N.LT.2)CALL ERROR
      N1=N-1
      M=N+1
      CALL SETM(1,0.0,ALFA,144)
C
C   GET FUEL FLOW IN EACH TUBE
C
      DO 5 J=1,N
    5 WF(J)=WFCUM(J+1)-WFCUM(J)
C
C   TUBE 1
C
      RFP1=WF(2)/WF(1)
      ALFA(2,1)=(W(1,1)-WHAT(1,1))/(W(1,1)*RFP1-W(1,2))
      IF (WHAT(M,1)/WTHAT(1).GE.0.01)
     1 ALFA(2,1)=(WTHAT(1)-WT(1))/(WT(2)-RFP1*WT(1))
      ALFA(2,1)=AMIN1(ALFA(2,1),1.0,1.0/RFP1)
      IF (ALFA(2,1).LT.0.0) ALFA(2,1)=0.0
      ALFA(1,2)=AMIN1(1.0,RFP1*ALFA(2,1))
      ALFA(1,1)=1.0-ALFA(1,2)
C
C   TUBES 2 THRU (N-1)
C
      IF(N.EQ.2)GO TO 15
      DO 10 J=2,N1
      RFP1=WF(J+1)/WF(J)
      RFM1=WF(J-1)/WF(J)
      ALFA(J+1,J)=(W(J,J)-WHAT(J,J)+(W(J,J-1)-RFM1*W(J,J))*ALFA(J-1,J))
     1/(RFP1*W(J,J)-W(J,J+1))
      IF (WHAT(M,J)/WTHAT(J).GE.0.01)
```

```
     1 ALFA(J+1,J)=(WTHAT(J)-WT(J)-(WT(J-1)-RFM1*WT(J))*ALFA(J-1,J))
     1 /(WT(J+1)-RFP1*WT(J))
      ATEMP=1.0-ALFA(J,J-1)
      ALFA(J+1,J)=AMIN1(ALFA(J+1,J),1.0,ATEMP/RFP1)
      IF (ALFA(J+1,J).LT.0.0) ALFA(J+1,J)=0.0
      ALFA(J,J+1)=AMIN1(ATEMP,RFP1*ALFA(J+1,J))
   10 ALFA(J,J)=AMAX1(0.0,ATEMP-ALFA(J,J+1))
C
C    TUBE N
C
   15 ALFA(N,N)=1.0-ALFA(N,N1)
C
C     AIR ENTRAINMENT
C
      ALFA(M,N)=1.0
C
C     FINAL CHECK FOR OUT-OF-BOUNDS ALFA'S
C
      ALFMIN= 0.
      ALFMAX= 1.
      DO 17 I=1,M
      DO 17 J=1,M
      ALFMIN= AMIN1(ALFMIN,ALFA(J,I))
   17 ALFMAX= AMAX1(ALFMAX,ALFA(J,I))
      IF((ALFMIN.GE.0. ) .AND. (ALFMAX.LE.1. ) ) GO TO 70
      WRITE (6,BADALF)
C     KILL THE MIXING ON THIS STEP
      CALL SETM(1,0.,ALFA,144)
      DO 18 J=1,M
      CALL MOVF(1,W(1,J),WHAT(1,J),12,1)
   18 ALFA(J,J)= 1.
   70 RETURN
C
      END
```

```
CAUTOO       AUTOMATIC ELIMINATION FROM ERROR CONSIDERATIONS
C            OF SPECIES WITH NON-REPRESENTATIVE ERRORS
      SUBROUTINE AUTO
C
C     AUTOMATIC ELIMINATION FROM ERROR CONSIDERATIONS OF SPECIES WITH
C     NON-REPRESENTATIVE ERRORS
C
      REAL MEDIAN
C
      DIMENSION ERROR(25),EE(25)
C
      COMMON/COND/DUM1(33),LS,LSP3,NEXT
      COMMON/PORF/DUM2(84),E(28)
      COMMON/SKIP/NEGL(25),I1,I2,IT
C
      EQUIVALENCE (EE(1),E(4))
C
      I2 = 0
      M = LS - I1
      IF (M .LE. 3) RETURN
C
      K = 0
      DO 3 I=1,LS
      IF (I1 .EQ. 0) GO TO 2
      DO 1 J=1,I1
      IF (I .EQ. NEGL(J)) GO TO 3
    1 CONTINUE
    2 K = K + 1
      ERROR(K) = EE(I)
    3 CONTINUE
C
      N = M/2 + 1
C
      DO 5 NN=1,N
      EMX = ERROR(NN)
      I = NN
      NI = NN + 1
      DO 4 II=NI,M
      IF (ERROR(II) .LE. EMX) GO TO 4
      EMX = ERROR(II)
      I = II
    4 CONTINUE
      ERROR(I) = ERROR(NN)
      ERROR(NN) = EMX
    5 CONTINUE
C
      MEDIAN = ERROR(N)
      CUTOFF = 15.*MEDIAN
C
      DO 8 I=1,LS
```

```
      IF (I1 .EQ. 0) GO TO 7                                            AUT00051
      DO 6 J=1,I1                                                       AUT00052
      IF (I .EQ. NEGL(J)) GO TO 8                                       AUT00053
    6 CONTINUE                                                          AUT00054
    7 IF (FE(I) .LE. CUTOFF) GO TO 8                                    AUT00055
      I2 = I2 + 1                                                       AUT00056
      K = I1 + I2                                                       AUT00057
      NEGL(K) = I                                                       AUT00058
    8 CONTINUE                                                          AUT00059
      IT = I1 + I2                                                      AUT00060
C                                                                       AUT00061
      RETURN                                                            AUT00062
      END                                                               AUT00063
```

```
CBLCK        -BLCK-     BLOCK DATA  ( SP/MW )                              BLCK0001
C                                                                          BLCK0002
      BLOCK DATA                                                           BLCK0003
C                                                                          BLCK0004
*     SPECIES NAMES AND MOLECULAR WEIGHTS                                  BLCK0005
C                                                                          BLCK0006
      COMMON / SNMW / ALSP(2,75), ALMW(75)                                 BLCK0007
C                                                                          BLCK0008
      DATA     ALSP/                                                       BLCK0009
     1 6HH     ,6H     ,6HO     ,6H     ,6HH2     ,6H     ,                BLCK0010
     2 6HO2    ,6H     ,6HOH    ,6H     ,6HH2O    ,6H     ,                BLCK0011
     3 6HCO    ,6H     ,6HCO2   ,6H     ,6HN2     ,6H     ,                BLCK0012
     4 6HAR    ,6H     ,6HNO    ,6H     ,6HN      ,6H     ,                BLCK0013
     5 6HHO2   ,6H     ,6HNO2   ,6H     ,6HNH3    ,6H     ,                BLCK0014
     6 6HFUEL  , 6H     , ,                                                BLCK0015
     7 118*0./                                                             BLCK0016
C                                                                          BLCK0017
      DATA     ALMW/                                                       BLCK0018
     '  1.008, 16.000,  2.016, 32.000, 17.008, 18.016,                     BLCK0019
     ' 28.010, 44.010, 28.016, 39.944, 30.008, 14.008,                     BLCK0020
     ' 33.008, 46.008, 17.032,  1.000,                                     BLCK0021
     4 59*0./                                                              BLCK0022
C                                                                          BLCK0023
      END                                                                  BLCK0024
```

```
CBLKRDA      -BLKRDA-      BLOCK DATA      ( REACTION DATA )                BLKRDA01
C                                                                           BLKRDA02
      BLOCK DATA                                                            BLKRDA03
C                                                                           BLKRDA04
*     REACTION DATA ( LSR - A - N - EACT - M - NEGL )                       BLKRDA05
C                                                                           BLKRDA06
      COMMON / LSLRSV / LS, LR                                              BLKRDA07
C                                                                           BLKRDA08
      COMMON / REAC / LSR(4,30), DUMREA(15)                                 BLKRDA09
C                                                                           BLKRDA10
      COMMON / RRAT / A(30), N(30), EACT(30), RDUM(30), M(25,30), DUMAL BLKRDA11
C                                                                           BLKRDA12
      REAL     M, N                                                         BLKRDA13
C                                                                           BLKRDA14
      COMMON / SKIP / NEGL(25), I1, I2, IT                                  BLKRDA15
C                                                                           BLKRDA16
      DATA     LS,LR/16,23/                                                 BLKRDA17
C                                                                           BLKRDA18
      DATA     I1,IT/7,7/                                                   BLKRDA19
C                                                                           BLKRDA20
      DATA (NEGL(I),I=1,7)/                                                 BLKRDA21
     * 1, 2, 10, 12, 13, 15, 16/                                            BLKRDA22
C                                                                           BLKRDA23
      DATA     ( ( LSR(I,J),I=1,4),J=1,19)/                                 BLKRDA24
     1  7,      5,      8,      1,                                          BLKRDA25
     2  7,      4,      8,      2,                                          BLKRDA26
     3  1,      5,      6,      0,                                          BLKRDA27
     4  0,      3,      1,      1,                                          BLKRDA28
     5  1,      2,      5,      0,                                          BLKRDA29
     6  2,      2,      4,      0,                                          BLKRDA30
     7  5,      3,      6,      1,                                          BLKRDA31
     8  1,      4,      5,      2,                                          BLKRDA32
     9  2,      3,      5,      1,                                          BLKRDA33
     J  6,      2,      5,      5,                                          BLKRDA34
     K  3,      4,      5,      5,                                          BLKRDA35
     L 12,     11,      9,      2,                                          BLKRDA36
     M 12,      4,     11,      2,                                          BLKRDA37
     N 12,      5,     11,      1,                                          BLKRDA38
     O  1,      4,     13,      0,                                          BLKRDA39
     P  1,     13,      5,      5,                                          BLKRDA40
     Q  5,     13,      6,      4,                                          BLKRDA41
     R  2,     13,      5,      4,                                          BLKRDA42
     S  1,     13,      3,      4/                                          BLKRDA43
C                                                                           BLKRDA44
      DATA     ( ( LSR(I,J),I=1,4),J=20,23)/                                BLKRDA45
     T 11,     13,     14,      5,                                          BLKRDA46
     U 14,      1,     11,      5,                                          BLKRDA47
     V 14,      2,     11,      4,                                          BLKRDA48
     W 11,      2,     14,      0/                                          BLKRDA49
C                                                                           BLKRDA50
```

```
      DATA      A/                                                      BLKRDA51
     * 1.1206042E+3, 2.50000E+12, 1.00000E+19, 1.120000E+13,            BLKRDA52
     * 5.30000E+15, 8.15000E+18, 2.30000E+13, 2.04000E+14,              BLKRDA53
     * 4.00000E+13, 8.40000E+13, 8.00000E+14, 3.10000E+13,              BLKRDA54
     * 6.40000E+09, 4.00000E+13, 1.00000E+15, 7.00000E+13,              BLKRDA55
     * 6.00000E+12, 6.00000E+12, 2.30000E+13, 1.00000E+13,              BLKRDA56
     * 7.20000E+14, 1.90000E+13, 9.40000E+14,                           BLKRDA57
     * 7*0./                                                            BLKRDA58
C                                                                       BLKRDA59
      DATA      N/                                                      BLKRDA60
     * 2.4863852, 0., -1.0000, 0.5000, 0., -1.2200,                     BLKRDA61
     * 6*0.,1.0,17*0./                                                  BLKRDA62
C                                                                       BLKRDA63
      DATA      EACT/                                                   BLKRDA64
     * -2574.7429, 48000.00, 0., 92600.00, -2780.00, 0.,                BLKRDA65
     * 5200.00, 16500.00, 10200.00, 18000.00, 45000.00, 334.00,         BLKRDA66
     * 6250.00, 0., -1300.00, 5*0., 1930.00, 1060.00, -1930.00,         BLKRDA67
     * 7*0./                                                            BLKRDA68
C                                                                       BLKRDA69
      DATA      ( ( M(I,J),I=1,16),J=1,9)/                              BLKRDA70
     1 1.,1.,1.,1.,1.,1.,1.,1.,                                         BLKRDA71
     * 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA72
     2 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA73
     * 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA74
     3 1.07, 1.07, 4.5, 4.5, 4.5, 18.0, 4.5, 8.0,                       BLKRDA75
     * 3.6, 1.0, 4.5, 1., 1., 1., 1., 1.,                               BLKRDA76
     4 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA77
     * 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA78
     5 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA79
     * 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA80
     6 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA81
     * 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA82
     7 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA83
     * 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA84
     8 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA85
     * 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA86
     9 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA87
     * 1., 1., 1., 1., 1., 1., 1., 1./                                  BLKRDA88
      DATA      ( ( M(I,J),I=1,16),J=10,18)/                            BLKRDA89
     1 1.,1.,1.,1.,1.,1.,1.,1.,                                         BLKRDA90
     * 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA91
     1 1.,1.,1.,1.,1.,1.,1.,1.,                                         BLKRDA92
     1 1.,1.,1.,1.,1.,1.,1.,1.,                                         BLKRDA93
     1 1.,1.,1.,1.,1.,1.,1.,1.,                                         BLKRDA94
     2 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA95
     1 1.,1.,1.,1.,1.,1.,1.,1.,                                         BLKRDA96
     3 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA97
     1 1.,1.,1.,1.,1.,1.,1.,1.,                                         BLKRDA98
     4 1., 1., 1., 1., 1., 1., 1., 1.,                                  BLKRDA99
     1 1., 1., 5.0, 2.0, 2.0, 32.5, 2.0, 7.5,                           BLKRDA00
```

```
     5 2.0, 1.0, 2.0, 1., 1., 1., 1., 1.,                          BLKRDA01
     1 1.,1.,1.,1.,1.,1.,1.,1.,                                     BLKRDA02
     6 1., 1., 1., 1., 1., 1., 1., 1.,                              BLKRDA03
     1 1.,1.,1.,1.,1.,1.,1.,1.,                                     BLKRDA04
     7 1., 1., 1., 1., 1., 1., 1., 1.,                              BLKRDA05
     1 1.,1.,1.,1.,1.,1.,1.,1.,                                     BLKRDA06
     8 1., 1., 1., 1., 1., 1., 1., 1./                              BLKRDA07
C                                                                   BLKRDA08
      DATA      ( ( M(I,J),I=1,16),J=19,23)/                        BLKRDA09
     1 1.,1.,1.,1.,1.,1.,1.,1.,                                     BLKRDA10
     9 1., 1., 1., 1., 1., 1., 1., 1.,                              BLKRDA11
     2 1., 1., 1., 1., 1., 1., 1., 1.,                              BLKRDA12
     * 1., 1., 1., 1., 1., 1., 1., 1.,                              BLKRDA13
     2 1., 1., 1., 1., 1., 1., 1., 1.,                              BLKRDA14
     1 1.,1.,1.,1.,1.,1.,1.,1.,                                     BLKRDA15
     2 1., 1., 1., 1., 1., 1., 1., 1.,                              BLKRDA16
     2 1., 1., 1., 1., 1., 1., 1., 1.,                              BLKRDA17
     2 0.75, 1., 0.75, 1., 1., 6.1, 1., 2.1,                        BLKRDA18
     3 1.4, 1., 1.4, 1., 1., 1., 1., 1./                            BLKRDA19
C                                                                   BLKRDA20
C                                                                   BLKRDA21
      END                                                           BLKRDA22
```

```
CBLKST        STOICHIOMETRIC COEFFICIENTS
      BLOCK  DATA
C
C*    STOICHIOMETRIC COEFFICIENTS FOR REACTIONS
C
      COMMON / SPEC / DUM1(110), STOIC(25,30), DUM2(750)
C
      DATA     ( ( STOIC(I,J),I=1,16),J=1,9)/
     1 1., 0., 0., 0., -1., 0., -1., 1.,
     *0., 0., 0., 0., 0., 0., 0., 0.,
     2 0., 1., 0., -1., 0., 0., -1., 1.,
     *0., 0., 0., 0., 0., 0., 0., 0.,
     3 -1., 0., 0., 0., -1., 1., 0., 0.,
     *0., 0., 0., 0., 0., 0., 0., 0.,
     4 2., 0., -1., 0., 0., 0., 0., 0.,
     *0., 0., 0., 0., 0., 0., 0., 0.,
     5 -1., -1., 0., 0., 1., 0., 0., 0.,
     *0., 0., 0., 0., 0., 0., 0., 0.,
     6 0., -2., 0., 1., 0., 0., 0., 0.,
     *0., 0., 0., 0., 0., 0., 0., 0.,
     7 1., 0., -1., 0., -1., 1., 0., 0.,
     *0., 0., 0., 0., 0., 0., 0., 0.,
     8 -1., 1., 0., -1., 1., 0., 0., 0.,
     *0., 0., 0., 0., 0., 0., 0., 0.,
     9 1., -1., -1., 0., 1., 0., 0., 0.,
     * 0., 0., 0., 0., 0., 0., 0., 0./
C
      DATA     ( ( STOIC(I,J),I=1,16),J=10,18)/
     1 0., -1., 0., 0., 2., -1., 0., 0.,
     *0., 0., 0., 0., 0., 0., 0., 0.,
     1 0., 0., -1., -1., 2., 0., 0., 0.,
     10., 0., 0., 0., 0., 0., 0., 0.,
     1 0., 1., 0., 0., 0., 0., 0., 0.,
     2 1., 0., -1., -1., 0., 0., 0., 0.,
     1 0., 1., 0., -1., 0., 0., 0., 0.,
     3 0., 0., 1., -1., 0., 0., 0., 0.,
     1 1., 0., 0., 0., -1., 0., 0., 0.,
     4 0., 0., 1., -1., 0., 0., 0., 0.,
     1 -1., 0., 0., -1., 0., 0., 0., 0.,
     5 0., 0., 0., 0., 1., 0., 0., 0.,
     1 -1., 0., 0., 0., 2., 0., 0., 0.,
     6 0., 0., 0., 0., -1., 0., 0., 0.,
     1 0., 0., 0., 1., -1., 1., 0., 0.,
     7 0., 0., 0., 0., -1., 0., 0., 0.,
     1 0., -1., 0., 1., 1., 0., 0., 0.,
     8 0., 0., 0., 0., -1., 0., 0., 0./
C
      DATA     ( ( STOIC(I,J),I=1,16),J=19,23)/
     1 -1., 0., 1., 1., 0., 0., 0., 0.,
     9 0., 0., 0., 0., -1., 0., 0., 0.,
```

```
     2 0., 0., 0., 0., 1., 0., 0., 0.,                               BLKST051
     * 0., 0., -1., 0., -1., 1., 0., 0.,                             BLKST052
     2 -1., 0., 0., 0., 1., 0., 0., 0.,                              BLKST053
     1 0., 0., 1., 0., 0., -1., 0., 0.,                              BLKST054
     2 0., -1., 0., 1., 0., 0., 0., 0.,                              BLKST055
     2 0., 0., 1., 0., 0., -1., 0., 0.,                              BLKST056
     2 0., -1., 0., 0., 0., 0., 0., 0.,                              BLKST057
     3 0., 0., -1., 0., 0., 1., 0., 0./                              BLKST058
C                                                                    BLKST059
      END                                                            BLKST060
```

```
CBLKTH      -BLKTH-      BLOCK DATA  ( THERMO )
C
      BLOCK DATA
C
*     THERMODYNAMIC DATA
C
C
      COMMON / TCOF / TC(7,2,25), TLOW, TMID, THI
      COMMON /COND  / DUMC1(33), LS , DUMC2(2)
C
      DATA        LS/16/
      DATA     TLOW, TMID, THI/300.0,1000.0,5000.0/
C
C
      DATA     ( ( ( TC(K,KK,I),K=1,7),KK=1,2),I=1,3)/
     *  0.25000000E 01,  0.              ,  0.              ,
     *  0.              ,  0.              ,  0.25981225E 05,
     * -0.46011763E 00,  0.25000000E 01,  0.              ,
     *  0.              ,  0.              ,  0.              ,
     *  0.25981225E 05, -0.46011762E 00,
     *  0.25420596E 01, -0.27550619E-04, -0.31028033E-08,
     *  0.45510674E-11, -0.43680515E-15,  0.29753241E 05,
     *  0.49203080E 01,  0.29464287E 01, -0.16381665E-02,
     *  0.24210316E-05, -0.16028432E-08,  0.38906964E-12,
     *  0.29670083E 05,  0.29639949E 01,
     *  0.31001901E 01,  0.51119464E-03,  0.52644210E-07,
     * -0.34909973E-10,  0.36945345E-14,  0.14181509E 03,
     * -0.19629421E 01,  0.30574451E 01,  0.26765200E-02,
     * -0.58099162E-05,  0.55210391E-08, -0.18122739E-11,
     *  0.30290764E 02, -0.22997056E 01/
C
      DATA     ( ( ( TC(K,KK,I),K=1,7),KK=1,2),I=4,6)/
     *  0.36219535E 01,  0.73618264E-03, -0.19652228E-06,
     *  0.36201558E-10, -0.28945627E-14, -0.15710570E 03,
     *  0.36150960E 01,  0.36255985E 01, -0.18782184E-02,
     *  0.70554544E-05, -0.67635137E-08,  0.21555993E-11,
     * -0.26457977E 01,  0.43052778E 01,
     *  0.29131230E 01,  0.95418248E-03, -0.19084325E-06,
     *  0.12730795E-10,  0.24803941E-15,  0.49967421E 04,
     *  0.54288735E 01,  0.38365518E 01, -0.10702014E-02,
     *  0.94849757E-06,  0.20843575E-09, -0.23384265E-12,
     *  0.47036168E 04,  0.49805456E 00,
     *  0.27167633E 01,  0.29451374E-02, -0.80224374E-06,
     *  0.10226682E-09, -0.48472145E-14, -0.28364192E 05,
     *  0.66305671E 01,  0.40701275E 01, -0.11084499E-02,
     *  0.41521180E-05, -0.29637404E-08,  0.80702103E-12,
     * -0.28738088E 05, -0.32270046E 00/
C
      DATA     ( ( ( TC(K,KK,I),K=1,7),KK=1,2),I=7,9)/
     *  0.29840696E 01,  0.14891390E-02, -0.57899684E-06,
```

```
     *  0.10364577E-09, -0.69353550E-14, -0.13595894E 05,                BLKTH051
     *  0.63479156E 01,  0.37100928E 01, -0.16190964E-02,                BLKTH052
     *  0.36923594E-05, -0.20319674E-08,  0.23953344E-12,                BLKTH053
     * -0.13706976E 05,  0.29555351E 01,                                 BLKTH054
     *  0.44608041E 01,  0.30981719E-02, -0.12392571E-05,                BLKTH055
     *  0.22741325E-09, -0.15525954E-13, -0.47789669E 05,                BLKTH056
     * -0.98635982E+00,  0.24007797E+01,  0.87350957E-02,                BLKTH057
     * -0.66070878E-05,  0.20021861E-08,  0.63274039E-15,                BLKTH058
     * -0.47205754E 05,  0.96951457E 01,                                 BLKTH059
     *  0.28963194E 01,  0.15154866E-02, -0.57235277E-06,                BLKTH060
     *  0.99807393E-10, -0.65223555E-14,  0.13750436E 03,                BLKTH061
     *  0.61615148E 01,  0.36748261E 01, -0.12081500E-02,                BLKTH062
     *  0.23240102E-05, -0.63217559E-09, -0.22577253E-12,                BLKTH063
     * -0.17792603E 02,  0.23580424E 01/                                 BLKTH064
C                                                                        BLKTH065
      DATA      ( ( ( TC(K,KK,I),K=1,7),KK=1,2),I=10,12)/                BLKTH066
     *  0.25000000E 01,  0.             ,  0.             ,              BLKTH067
     *  0.             ,  0.             ,  0.             ,              BLKTH068
     *  0.43660006E 01,  0.25000000E 01,  0.             ,               BLKTH069
     *  0.             ,  0.             ,  0.             ,              BLKTH070
     *  0.45776367E-04,  0.43660006E+01,                                 BLKTH071
     *  0.31890000E 01,  0.13382281E-02, -0.52899318E-06,                BLKTH072
     *  0.95919332E-10, -0.64847932E-14,  0.10872450E 05,                BLKTH073
     *  0.67458126E 01,  0.40459521E 01, -0.34181783E-02,                BLKTH074
     *  0.79819190E-05, -0.61139316E-08,  0.15919076E-11,                BLKTH075
     *  0.10789515E 05,  0.29974988E 01,                                 BLKTH076
     *  0.24502682E 01,  0.10661458E-03, -0.74653373E-07,                BLKTH077
     *  0.18796524E-10, -0.10259839E-14,  0.56637723E 05,                BLKTH078
     *  0.44875581E 01,  0.25030714E 01, -0.21800181E-04,                BLKTH079
     *  0.54205287E-07, -0.56475602E-10,  0.20999044E-13,                BLKTH080
     *  0.56620587E 05,  0.41675764E 01/                                 BLKTH081
C                                                                        BLKTH082
      DATA      ( ( ( TC(K,KK,I),K=1,7),KK=1,2),I=13,15)/                BLKTH083
     *  0.37866280E 01,  0.27885404E-02, -0.10168708E-05,                BLKTH084
     *  0.17183946E-09, -0.11021852E-13,  0.27433246E 04,                BLKTH085
     *  0.48147611E 01,  0.35094850E 01,  0.11499670E-02,                BLKTH086
     *  0.58784259E-05, -0.77795519E-08,  0.29607883E-11,                BLKTH087
     *  0.29348076E 04,  0.68276325E 01,                                 BLKTH088
     *  0.46240771E 01,  0.25260332E-02, -0.10609498E-05,                BLKTH089
     *  0.19879239E-09, -0.13799384E-13,  0.38565499E 04,                BLKTH090
     *  0.13324138E 01,  0.34589236E 01,  0.20647064E-02,                BLKTH091
     *  0.66866067E-05, -0.95556725E-08,  0.36195881E-11,                BLKTH092
     *  0.43817864E 04,  0.83116983E 01,                                 BLKTH093
     *  0.24165177E 01,  0.61871211E-02, -0.21785136E-05,                BLKTH094
     *  0.37599090E-09, -0.24448856E-13, -0.44242413E 04,                BLKTH095
     *  0.77034482E 01,  0.35912768E 01,  0.49388668E-03,                BLKTH096
     *  0.83449322E-05, -0.83833385E-08,  0.27299092E-11,                BLKTH097
     * -0.46212379E 04,  0.22520966E 01/                                 BLKTH098
C                                                                        BLKTH099
      DATA      ( ( ( TC(K,KK,I),K=1,7),KK=1,2),I=16,16)/                BLKTH100
```

```
     1 3.4552152,1.1491803E-2,-4.3651750E-6,                          BLKTH101
     6 7.6155095E-10,-5.0123200E-14,6.7694946E3,2.6987959,            BLKTH102
     * 1.4256821,1.1383140E-2,7.9890006E-6,-1.6253679E-8,             BLKTH103
     * 6.7491256E-12,7.6292582E3,1.4621819E1/                         BLKTH104
C                                                                     BLKTH105
      END                                                             BLKTH106
```

```
*BLOCKT          BLOCK DATA FOR TURBULENT PARAMETERS
      BLOCK DATA
      REAL MUL,MUEFF,KCP,MUREF,MACH
      COMMON /CPROP/ CT(10)
      COMMON /CPROP2/ CTP , CTS , CTM
C*
      COMMON /PROPJT/
     * P         , PRL      , PRT       , RGAS     , SC       ,
     * TREF      , MUREF    , MACH      , XLC      ,
     * REFL      , C        , CHI       , RNORM    ,
     * RHO(200)    , MUL(200)    , KCP(200)    ,
     * MUEFF(200) , XLN(200)    , DK(200)     , RETURB(200)
C*
C*
      DATA
     * C/2.59/   , PRL/.72/  , PRT/1./  , RGAS/53.34/ ,
     * TREF/0./  , MUREF/0./  , SC/216./  ,
     * CHI/.586/ , RNORM/110./
      DATA CT/.23,.38,.23,.3,.38,1.4,.43,.1875,.9,0./
      DATA CTP/.175/ , CTM/.23/ , CTS/.23/
C*
      END
```

```
CCAROL        FUEL-AIR RATIO AND EMISSION INDICES FROM CAROL ANALYSIS
C             OF GAS SAMPLE
      SUBROUTINE CAROL(N,WAR,RHC,RCO,RCO2,RNOX,FAR,EIHC,EICO,EINOX)
      REAL N,I,MD,MT,MWTF,MWT
      XM(X,Y)=(X*(F-C)-Y*(I-C))/((G-A)*(E-B)-(D-A)*(H-B))
C
      A=1.0-(0.1-0.25*N)*RHC
      B=-0.503101*(RCO+RH2(RCO))
      C=-(0.03452247+WAR/18.016)
      D=(0.1+0.25*N)*RHC
      E=0.503101*(RCO+3.0*RH2(RCO))-1.0
      F=0.03452247
      G=0.25*N*RHC
      H=0.2515505*N*(RCO+RCO2)
      I=-2.5891851E-6*N
C
      MD=-XM(G-A,D-A)
      MT=XM(H-B,F-B)
      MWTF=120.1+10.08*N
      FAR=0.4*MWTF/N*(A*MT+B*MD+C)
      GPF=(1.0+FAR+WAR)/FAR
      MWT=GPF*FAR/MT
C
      EIHC=RHC/10.0*MWTF/MWT*GPF
      EICO=RCO*1.006202*MD/MT*28.01/MWT*GPF
      EINOX=RNOX*46.008/MWT*GPF
C
      RETURN
      END
```

```
CCASMM      CHOOSE STEP FORMULA, SET UP AUGMENTED MATRIX, INCREMENTS
      SUBROUTINE CASM
C
C     CHOOSE  (1) INITIAL STEP (RESTART) FORMULA
C             (2) GENERAL STEP FORMULA
C
C     SET UP AUGMENTED MATRIX
C
C     COMPUTE INCREMENTS
C
      DOUBLE PRECISION A
C
      REAL IVAR
C
      COMMON/OPTS/VERSI,TIMEV,VERSA,AREAV,FLIM,ICON,RHOCON,IPRCOD
      COMMON/COND/DVAR,AREA,MDOT,P,IVAR,Y(28),LS,LSP3,NEXT
      COMMON/DERN/F(28),ALPHA(28),BETA(28,28)
      COMMON/MATX/A(28,29)
      DIMENSION QK(28),RK(28)
      EQUIVALENCE (VN1,Y(1))
C
      DATA EPS/0.0001/
C
      ENTRY CASI (HN,QK,HN1,RK)
C
C
C     INITIAL STEP OR RESTART
      INGEN = 1
      JK = 0
      NEGS=0
   10 F1 = 0.
      F2 = HN1
      F3 = HN1/2.
      F4 = F3
      GO TO 2
C
      ENTRY CASG (HN,QK,HN1,RK)
C     GENERAL STEP
      INGEN = 2
      JK = 0
      NEGS=0
   20 F1 = HN1*HN1/((2.*HN1 + HN)*HN)
      F2 = HN1*(HN1 + HN)/(2.*HN1 + HN)
      F3 = HN1
      F4 = F2
C
    2 LSP4 = LSP3 + 1
C
      DO 4 I=1,LSP3
      DO 3 J=1,LSP3
```

```
    3 A(I,J) = -F4*BETA(I,J)                                          CASMM051
      A(I,I) = 1.D0 + A(I,I)                                          CASMM052
      IF (DABS(A(I,I)) .GE. EPS) GO TO 4                              CASMM053
      IF (JK .GE. 3) GO TO 4                                          CASMM054
      JK = JK + 1                                                     CASMM055
      HN1 = 2.*((1. - EPS)/BETA(I,I))                                 CASMM056
      GO TO (10,20),INGEN                                             CASMM057
    4 A(I,LSP4) = F1*QK(I) + F2*(F(I) + ALPHA(I)*F3)                  CASMM058
C                                                                     CASMM059
      CALL LESV (RK)                                                  CASMM060
      DO 7 I=1,LSP3                                                   CASMM061
      IF (Y(I)+RK(I) .GE. 0.0) GO TO 7                                CASMM062
      NEGS = NEGS+1                                                   CASMM063
      IF (NEGS .GT. 4) GO TO 7                                        CASMM064
      HN1 = 0.5*HN1                                                   CASMM065
      GO TO (10,20),INGEN                                             CASMM066
    7 CONTINUE                                                        CASMM067
      VN = Y(1)                                                       CASMM068
      DO 5 I=1,LSP3                                                   CASMM069
    5 Y(I) = Y(I) + RK(I)                                             CASMM070
      IVAR = IVAR + HN1                                               CASMM071
C                                                                     CASMM072
      IF (VERSI .EQ. TIMEV) GO TO 6                                   CASMM073
      DVAR = DVAR + 2.*HN1/(VN + VN1)                                 CASMM074
      RETURN                                                          CASMM075
    6 DVAR = DVAR + (VN + VN1)/2.*HN1                                 CASMM076
      RETURN                                                          CASMM077
C                                                                     CASMM078
      END                                                             CASMM079
```

```
CCFILE         MERGE OR SPLIT FILES
      SUBROUTINE MAIN
C
      COMMON /FILK  / CSC
      COMMON /CPRJE1/ NPTS,RAD(12),TS(12),UJ(12),SPV(12),
     * MWT(12),CP(12),FUEL(12),SPALDG(12),TKE(12),OTHER(36),
     * TITLE(20),PRINT(20),GEX,N,HF,WAR,T2,BETA,T25,FARS,EINO2C,
     * P0,RCO(11),RCO2(11),RHC(11),RNOX(11),PT(11),PS(11),
     * BLOC(11),OCO(11),RICH(24),FUUL(48),ENTH(48),CONC(768),
     * HCINCP(24),OTHER2(192)
      COMMON /CGJET / GJET(200)
C
C
      COMMON /CJMIX1/ NAME(10),TITLE1(10),IDENT(10),ADDRES(10),
     * IDENT1(10),TWODT(9),BITS,ERK,GC,GCJ,FOOT,DIAJ,MJET,TJET,
     * PTJET,VJET,TIJET,EJET,PE,VE,ME,TIF,TE,AXI,NJ,NM,UE,
     * MIXPRE,XLC,FLOWJ,MERGE,NV,CON1,CT1,CT2,CT3,CT4,CT5,CT6,
     * CT7,CT8,CT9,CTP,CTS,CTM,GAM,RG,PR,PRT,SC,TREF,MUREF,
     * SP,SV,SLEN,DPRIN,PLOT,C6,MIX,CF,MAXIT,TOL,SUPB,
     * X(100),XPRV(100),B(100),UC(100),TC(100),TIC(100),
     * PTC(100),WJ(100),YJ(100),TTC(100),YSONIC(100),YCB(100),
     * XD(100),RD(100),YR(100),YCD(100),PD(100),WV(100),
     * MA2(100),VE2(100),TE2(100),NXTA,I,NC,CNAME(12),ALJ(12),
     * ALJD(12),ALE(12),SCM(12),CPC(36),DIFF
C
      COMMON /CJMIX2/ SUPD,SUPSTP,CORE,CORSTP,MER,MERSTP,NPD,
     * PSI(200),Y(200),UD(200),ED(200),TJD(200),RHO(200),
     * XLV(200),U(200),TOT(200),XMACH(200),PTOT(200),TTD(200),
     * PTD(200),MOLF1(100),MOLF2(100),MOLF3(100),MOLF4(100),
     * MOLF5(100),MOLF6(100),MOLF7(100),MOLF8(100),MOLF9(100),
     * MOLF10(100),MOLF11(100),MOLF12(100),T(200)
C
      COMMON /CSPALD/ SY(50),RADS(50),GS(50),JEND
C
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
      LOGICAL         ERR,ERRMAJ,INERR,PRERR
      LOGICAL MERGE , SPLIT
      NAMELIST /A/ MERGE,SPLIT,NF
C
C   MERGE = T        MERGE FILES
C   SPLIT = T        SPLIT OFF FILES
C   NF                FILE PRESENT
C         = 1         PREJET
C         = 2         PREJET,JETMIX
C         = 3         PREJET,JETMIX,SPALDG
C
      CALL FLGERR(5,ERR)
    1 MERGE = .FALSE.
      SPLIT = .FALSE.
      NF    = 0
```

```
      READ (5,A)
      IF( ERR ) RETURN
      IF( SPLIT ) GO TO 500
C
    5 WRITE (6,7)
    7 FORMAT(1H1,10X,17H** FILE MERGE **/10X,14HDATA ON FILE 4//
     * 10X,6HPREJET)
C
   10 WRITE (4) NF
C
C
C*    SECTION TO MERGE FILES ( JOB TERMINATION )
C
C*      PREJET **
C
   12 READ (1)
     * DIAJ,MJET,TIJET,VJET,PE,TF,TIE,VE,GFX,RG,PR,PRT,SC,TREF,
     * MUREF,DIFF,NC,CNAME,ALE,SCM,CPC,NJ,NM,CT1,CT2,CT3,CT4,CT5,
     * CT6,CT7,CT8,GJET,Y,UD,THD,TID,ED,
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,
     * NPTS,RAD,TS,UJ,SPV,MWT,CP,FUEL,SPALDG,TKE,OTHER,
     * TITLE,PRINT,N,HF,WAR,T2,BETA,T25,FARS,EINO2C,PO,
     * RCO,RCO2,RHC,RNOX,PT,PS,BLOC,QCO,
     * RICH,HCINCP,FUUL,ENTH,CONC,OTHER2
     * ,GEX
   14 WRITE (4)
     * DIAJ,MJET,TIJET,VJET,PE,TE,TIE,VE,GFX,RG,PR,PRT,SC,TREF,
     * MURFF,DIFF,NC,CNAME,ALE,SCM,CPC,NJ,NM,CT1,CT2,CT3,CT4,CT5,
     * CT6,CT7,CT8,GJET,Y,UD,THD,TID,ED,
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,
     * NPTS,RAD,TS,UJ,SPV,MWT,CP,FUEL,SPALDG,TKE,OTHER,
     * TITLE,PRINT,N,HF,WAR,T2,BETA,T25,FARS,EINO2C,PO,
     * RCO,RCO2,RHC,RNOX,PT,PS,BLOC,QCO,
     * RICH,HCINCP,FUUL,ENTH,CONC,OTHER2
     * ,GEX
      IF( NF.EQ.1 ) GO TO 1000
C
C*    JETMIX **
C
   20 WRITE (6,21)
   21 FORMAT(10X,6HJETMIX,18X,2HXX)
   23 READ (2) KXX1,KREC,
     * NAME,TITLE1,IDENT,ADDRES,IDENT1,TWODT,BITS,FRK,GC,GCJ,
     * FOOT,DIAJ,MJET,TJET,PTJET,VJET,TIJET,EJET,PE,VE,ME,TII..TE,
     * AXI,NJ,NM,UE,MIXPRF,XLC,FLOWJ,MERGE,NV,CON1,
     * CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,CT9,CTP,CTS,CTM,
     * GAM,RG,PR,PRT,SC,TREF,MUREF,SP,SV,SLEN,DPRIN,PLOT,C6,
     * MIX,CF,MAXIT,TOL,SUPH,
```

```
     * X,XPRN,B,UC,TC,TIC,PTC,WJ,YJ,TTC,YSONIC,                      CFILE101
     * YCB,XD,RD,YR,YCD,PD,WV,MA2,VE2,TE2,NXTA,I,                     CFILE102
     * NC,CNAME,ALJ,ALJO,ALE,SCM,CPC,DIFF,CSC                         CFILE103
       WRITE (4) KXX1,KREC,                                            CFILE104
     * NAME,TITLE1,IDENT,ADDRES,IDENT1,TWODT,BITS,ERK,GC,GCJ,         CFILE105
     * FOOT,DIAJ,MJET,TJET,PTJET,VJET,TIJET,EJET,PE,VE,ME,TIE,TE,     CFILE106
     * AXI,NJ,NM,UE,MIXPRE,XLC,FLOWJ,MERGE,NV,CON1,                   CFILE107
     * CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,CT9,CTP,CTS,CTM,               CFILE108
     * GAM,RG,PR,PRT,SC,TREF,MUREF,SP,SV,SLEN,DPRIN,PLOT,C6,          CFILE109
     * MIX,CE,MAXIT,IDL,SUPB,                                          CFILE110
     * X,XPRN,B,UC,TC,TIC,PTC,WJ,YJ,TTC,YSONIC,                       CFILE111
     * YCB,XD,RD,YR,YCD,PD,WV,MA2,VE2,TE2,NXTA,I,                     CFILE112
     * NC,CNAME,ALJ,ALJO,ALE,SCM,CPC,DIFF,CSC                         CFILE113
       DO 25 IJ=1,KREC                                                 CFILE114
       READ (2) JREC,KXX,KREG,                                         CFILE115
     * SUPD,SUPSTP,CORE,CORSTP,MER,MERSIP,NPD,                        CFILE116
     * PSI,Y,UD,THD,ED,TID,RHO,XLN,U,T,TOT,XMACH,                     CFILE117
     * PTOT,TID,PTD,                                                   CFILE118
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,                           CFILE119
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,J                       CFILE120
       WRITE (4) JREC,KXX,KREG,                                        CFILE121
     * SUPD,SUPSTP,CORE,CORSTP,MER,MERSIP,NPD,                        CFILE122
     * PSI,Y,UD,THD,ED,TID,RHO,XLN,U,T,TOT,XMACH,                     CFILE123
     * PTOT,TID,PTD,                                                   CFILE124
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,                           CFILE125
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,J                       CFILE126
       XKXX  = FLOAT(KXX)/CSC                                          CFILE127
       WRITE (6,24) XKXX                                               CFILE128
    24 FORMAT(10X,6HJETMIX,F20.3)                                      CFILE129
    25 CONTINUE                                                        CFILE130
C                                                                      CFILE131
       IF( NF.EQ.2 ) GO TO 1000                                        CFILE132
C                                                                      CFILE133
C*     SPALDG **                                                       CFILE134
C                                                                      CFILE135
    30 DO 35 IJ=1,KREC                                                 CFILE136
       READ (3) KXX,SY,RADS,GS,JEND                                    CFILE137
       WRITE (4) KXX,SY,RADS,GS,JEND                                   CFILE138
       XKXX  = FLOAT(KXX)/CSC                                          CFILE139
       WRITE (6,34) XKXX                                               CFILE140
    34 FORMAT(10X,6HSPALDG,F20.3)                                      CFILE141
    35 CONTINUE                                                        CFILE142
       GO TO 1000                                                      CFILE143
C                                                                      CFILE144
C*     SECTION TO SPLIT FILES ( JOB INITIATION )                       CFILE145
C                                                                      CFILE146
   500 READ (7) NF                                                     CFILE147
       WRITE (6,605)                                                   CFILE148
       READ (7)                                                        CFILE149
     * DIAJ,MJET,TIJET,VJET,PE,TE,TIE,VE,GEX,RG,PR,PRT,SC,TREF,       CFILE150
```

```
     * MURFF,DIFF,NC,CNAME,ALE,SCM,CPC,NJ,NM,CT1,CT2,CT3,CT4,CT5,
     * CT6,CT7,CT8,GJET,Y,UD,THD,TID,ED,
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,
     * NPTS,RAD,TS,UJ,SPV,MWT,CP,FUEL,SPALDG,TKE,OTHER,
     * TITLE,PRINT,N,HF,WAR,T2,BETA,T25,FAR5,EINO2C,PO,
     * RCO,RCO2,RHC,RNOX,PT,PS,BLOC,QCO,
     * RICH,HCINCP,FUUL,ENTH,CONC,OTHER2
     * ,GEX
 504 WRITE (1)
     * DIAJ,MJET,TIJET,VJET,PE,TE,TIE,VE,GEX,RG,PR,PRT,SC,TREF,
     * MUREF,DIFF,NC,CNAME,ALE,SCM,CPC,NJ,NM,CT1,CT2,CT3,CT4,CT5,
     * CT6,CT7,CT8,GJET,Y,UD,THD,TID,ED,
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,
     * NPTS,RAD,TS,UJ,SPV,MWT,CP,FUEL,SPALDG,TKE,OTHER,
     * TITLE,PRINT,N,HF,WAR,T2,BETA,T25,FAR5,EINO2C,PO,
     * RCO,RCO2,RHC,RNOX,PT,PS,BLOC,QCO,
     * RICH,HCINCP,FUUL,ENTH,CONC,OTHER2
     * ,GEX
      WRITE (6,600)
      IF( NF.EQ.1 ) GO TO 1000
C
 520 READ (7) KXX1,KREC,
     * NAME,TITLE1,IDENT,ADDRES,IDENT1,TWODT,BITS,ERK,GC,GCJ,
     * FOOT,DIAJ,MJET,TJET,PTJET,VJET,TIJET,EJET,PE,VE,ME,TIE,TE,
     * AXI,NJ,NM,UE,MIXPRE,XLC,FLOWJ,MERGE,NV,CON1,
     * CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,CT9,CTP,CTS,CTM,
     * GAM,RG,PR,PRT,SC,TREF,MURFF,SP,SV,SLEN,DPRIN,PLOT,C6,
     * MIX,CF,MAXIT,TOL,SUPB,
     * X,XPRN,B,UC,TC,TIC,PTC,WJ,YJ,TTC,YSONIC,
     * YCB,XD,RD,YR,YCD,PD,WV,MA2,VE2,TE2,NXTA,I,
     * NC,CNAME,ALJ,ALJO,ALE,SCM,CPC,DIFF,CSC
      WRITE (2) KXX1,KREC,
     * NAME,TITLE1,IDENT,ADDRES,IDENT1,TWODT,BITS,ERK,GC,GCJ,
     * FOOT,DIAJ,MJET,TJET,PTJET,VJET,TIJET,EJET,PE,VE,ME,TIE,TE,
     * AXI,NJ,NM,UE,MIXPRE,XLC,FLOWJ,MERGE,NV,CON1,
     * CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,CT9,CTP,CTS,CTM,
     * GAM,RG,PR,PRT,SC,TREF,MUREF,SP,SV,SLEN,DPRIN,PLOT,C6,
     * MIX,CF,MAXIT,TOL,SUPB,
     * X,XPRN,B,UC,TC,TIC,PTC,WJ,YJ,TTC,YSONIC,
     * YCB,XD,RD,YR,YCD,PD,WV,MA2,VE2,TE2,NXTA,I,
     * NC,CNAME,ALJ,ALJO,ALE,SCM,CPC,DIFF,CSC
      DO 525 IJ=1,KREC
      READ (7) JREC,KXX,KREG,
     * SUPD,SUPSTP,CORE,CORSTP,MER,MERSTP,NPD,
     * PSI,Y,UD,THD,ED,TID,RHO,XLN,U,T,TOT,XMACH,
     * PTOT,TTD,PTD,
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,J
```

```
      WRITE (2) JREC,KXX,KREG,                                          CFILE201
     * SUPD,SUPSTP,CORE,CORSTP,MFR,MFRSTP,NPD,                          CFILE202
     * PSI,Y,UD,TMD,ED,TID,RHO,XLN,J,T,TOT,XMACH,                       CFILE203
     * PTOT,TTD,PTD,                                                    CFILE204
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,                             CFILE205
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,J                         CFILE206
  525 CONTINUE                                                          CFILE207
      WRITE (6,601)                                                     CFILE208
C                                                                       CFILE209
      IF( NF.EQ.2 ) GO TO 1000                                          CFILE210
C                                                                       CFILE211
  530 DO 535 IJ=1,KREC                                                  CFILE212
      READ (7) KXX,SY,RADS,GS,JEND                                      CFILE213
      WRITE (3) KXX,SY,RADS,GS,JEND                                     CFILE214
  535 CONTINUE                                                          CFILE215
      WRITE (6,602)                                                     CFILE216
  600 FORMAT(10X,22H**PREJET FILE RESTORED,10X,12HFILE CODE   1/)       CFILE217
  601 FORMAT(10X,22H**JETMIX FILE RESTORED,10X,12HFILE CODE   2/)       CFILE218
  602 FORMAT(10X,22H**SPALDG FILE RESTORED,10X,12HFILE CODE   3/)       CFILE219
  605 FORMAT(1H1,10X,20H**FILE RESTORATION**//)                         CFILE220
 1000 RETURN                                                            CFILE221
      END                                                               CFILE222
```

```
CCOCO2B       FINITE RATE CO/CO2 CHEMISTRY
      SUBROUTINE COCO2B( P,DTIME )
      COMMON /PSEQ  / FOA,BETS,TP,X(16),DHQDMW,TEQ,BEQ,XMWT,HNEQ
      EQUIVALENCE      (X1,X(1)),(X2,X(2)),(X3,X(3)),(X4,X(4)),
     *                 (X5,X(5)),(X6,X(6)),(X7,X(7)),(X8,X(8)),
     *                 (X9,X(9)),(X10,X(10)),(X11,X(11))
      COMMON /GHSC  /  F(25),HH(25),SR(25),CPZ(25),DCPR(25)
      COMMON /CPRINT/ PDUM(20)
      COMMON /COREQN/ COREQ
      LOGICAL          COREQ
      COMMON /COLIMT/ XCOLIM
      COMMON /PSEQX/ HQC,HUM,CO2AIR,MAIR,FS,FUELMW
      REAL                              MAIR
      REAL         K10
      DATA C2/.01603286/,A10,E10/6.6E+11,1030./,TIME/0./
C
C        REACTION      CO+OH = CO2+H
C
      TK     = TP/1.8
    1 EQ10   = EXP(F(7)+F(5)-F(8)-F(1))
      IF( COREQ ) GO TO 2
      K10    = A10*EXP(-E10/(1.98596*TK))
      GO TO 3
    2 IF( TP.LT.2250. ) GO TO 21
      K10=1.3E+12*EXP(-5428.629857/(1.98596*TK))
      GO TO 3
   21 XNUM   = 1.342635076E+11*EXP(-704.02446/(1.98596*TK))
      XDEN   = 1.-78.5243789*EXP(-13752.289 /(1.98596*TK))
      K10    = XNUM/XDEN
    3 XCOI   = X7
      IF(X1.EQ.0.0 .AND. X5.EQ.0.0)GO TO 20
      RHOM   = 144.*P*C2/(1545.32*TP)
      TIMEK  = DTIME
      TIME   = TIME+TIMEK
      TERM1  = (X1+2.*X3+X5+2.*X6)/HQC
      A      = -RHOM*K10*(X5+X1/EQ10)
      B      = RHOM*K10*X1/EQ10*TERM1
      BQA    = B/A
   10 X7     = (XCOI+BQA)*EXP(A*TIMEK)-BQA
      IF( X7.LE.XCOLIM ) X7=XCOLIM
      DXCO   = XCOI-X7
      X8     = X8+DXCO
   20 RETURN
      END
```

```
CCOEFF        EVALUATES COEFFICIENTS FOR -G- EQUATION
C
      SUBROUTINE COEFF(K)
C
      COMMON /CENDS/ JSTART,JEND
      COMMON /CPROLI/ G(50,1),ALPHA(1,1),BETA(50,1),GAMM(50,1),
     1  DELTA(50,1)
      COMMON /CJETOT/ GCJ,DTAJ,VJET,EJET,NXTA
      COMMON /CCONST/ CONST1,CONST2,CONST3,CONST4
      COMMON /CSLDAT/ SY(50),RAD(50),VEL(50),TKE(50),DEN(50),TLS(50),
     1  FAR(50)
C
  110 DO 200 J=JSTART,JEND
C*  CALCULATION OF TURBULENT VISCOSITY -- LBM/FT-SEC
      ZMUT=CONST1*DEN(J)*TLS(J)*SQRT(TKE(J))
      IF (J.GT.1) GO TO 120
C*  SYMMETRY FORCES DFODSY TO BE ZERO ON AXIS
      DFODSY=0.
      GO TO 130
  120 JP1=J+1
      JM1=J-1
C*  AT OUTER EDGE OF JET, USE ONE-SIDED DERIVATIVE
      IF (J.EQ.JEND) JP1=J
      DFODSY=(FAR(JP1)-FAR(JM1))/(SY(JP1)-SY(JM1))
  130 CONTINUE
C*  CALCULATE COEFFICIENTS OF THE -G- EQUATION
      BETA(J,K)=CONST4*ZMUT*DEN(J)*VEL(J)*RAD(J)*RAD(J)
      GAMM(J,K)=CONST2*ZMUT*DEN(J)*VEL(J)*RAD(J)*RAD(J)*DFODSY*DFODSY
      IF (VEL(J).LT.(1.E-10)) GO TO 140
      DELTA(J,K)=-CONST3*ZMUT/(DEN(J)*VEL(J)*TLS(J)*TLS(J))
      GO TO 200
  140 DELTA(J,K)=0.0
  200 CONTINUE
      RETURN
      END
```

```
*CTABPR          BLOCK DATA FOR TABPRT                            'CTABPR'     CTABPR01
      BLOCK DATA                                                               CTABPR02
      COMMON /CTABPR/ IITAB                                                    CTABPR03
      DATA IITAB/1/                                                            CTABPR04
      END                                                                      CTABPR05
```

```
CCUBSS      COEFFICIENTS FOR CUBIC SPLINE INTERPOLATION AND DIFF'N
      SUBROUTINE CUBS (X,Y,N)
C
      DIMENSION X(N),Y(N)
      DIMENSION A(40),B(40),C(40)
      DIMENSION S(40),T(40),U(40)
C
      COMMON/AFUN/C1,C2,C3,C4,IIPSZ,ELM,ETA,DIAM,VISC,BETA
C
      EQUIVALENCE (A(1),U(1)),(B(1),T(1)),(C(1),S(1))
C
      F(X) = ((A1*X + A2)*X + A3)*X + A4
      DF(X) = (3.*A1*X + 2.*A2)*X + A3
      D2F(X) = 6.*A1*X + 2.*A2
C
      G(A) = 1./(1. - A**ETA)
      DG(B) = ETA/ELM*TERM**(ETA-1.)*B*B
      D2G(C,D,E) = C*(ETA - 1. + 2.*ETA*TERM**ETA*D)/E
C
C     FROM INPUT TABLE COMPUTE COEFFICIENTS FOR CUBIC SPLINE INTERPOLATIO
C     AND DIFFERENTIATION
      CON1 = .33333333
      CON2 = .16666667
      DXI = X(2) - X(1)
      DYI = Y(2) - Y(1)
      DI = DYI/DXI
      S(1) = CON1*DXI
      T(1) = CON2*DXI
      U(1) = DI - (((Y(3)-Y(1))/(X(3)-X(1))) + DI)/2.
      NM = N-1
      DO 2 I=2,NM
      DXIM = DXI
      DYIM = DYI
      DIM = DI
      DXI = X(I+1) - X(I)
      DYI = Y(I+1) - Y(I)
      DI = DYI/DXI
      S(I) = CON1*(DXIM + DXI)
      T(I) = CON2*DXI
    2 U(I) = DI - DIM
      S(N) = CON1*DXI
      U(N) = (DI + (DYI+DYIM)/(DXI+DXIM))/2. - DI
C
      DO 3 I=1,NM
      TT = T(I)
      T(I) = T(I)/S(I)
      U(I) = U(I)/S(I)
      S(I+1) = S(I+1) - TT*T(I)
    3 U(I+1) = U(I+1) - TT*U(I)
      U(N) = U(N)/S(N)
```

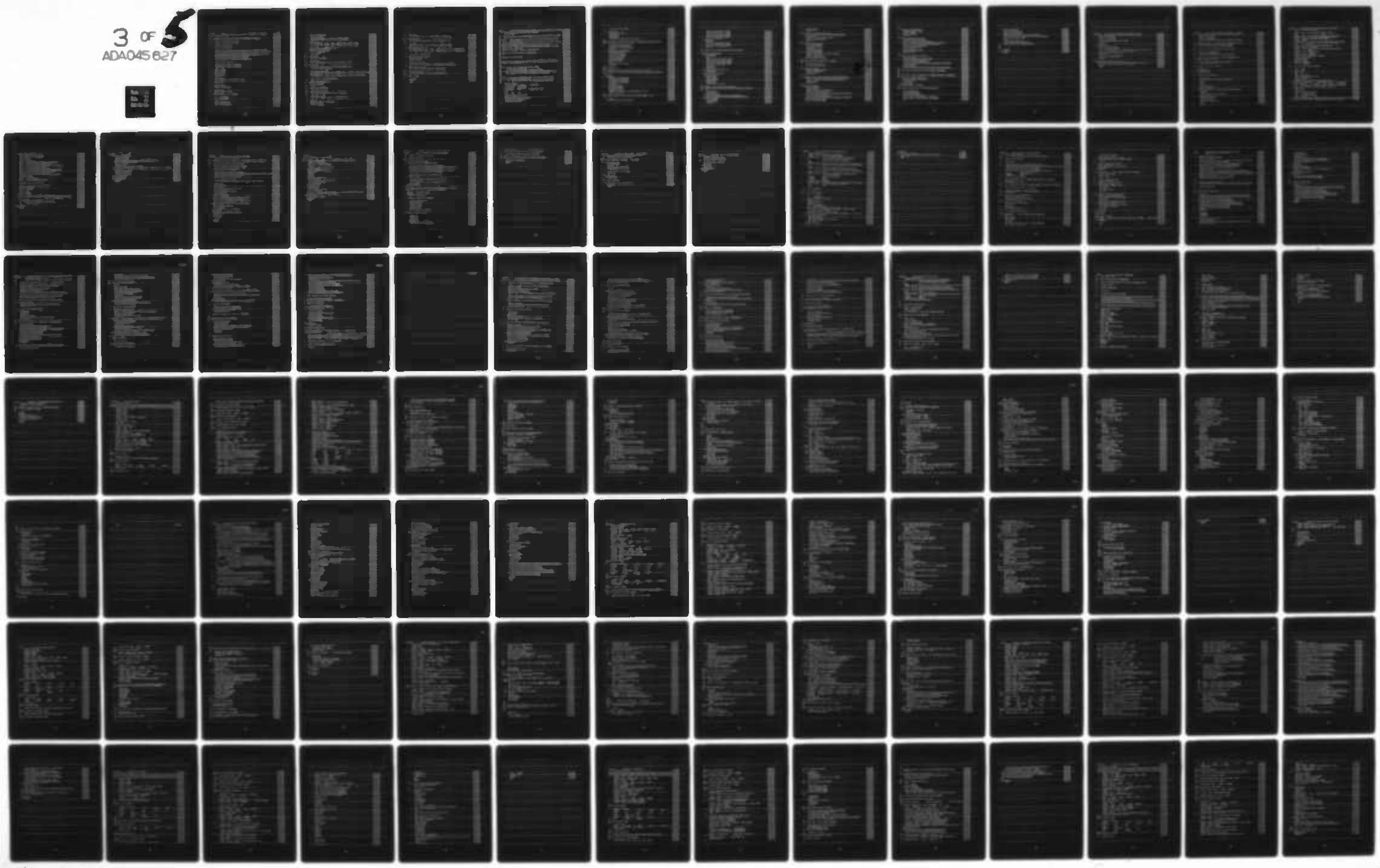
```
C
      A(N) = U(N)
      DO 4 J=1,NM
      I = N-J
    4 A(I) = U(I) - T(I)*A(I+1)
C
      DO 5 I=1,NM
      DXI = X(I+1) - X(I)
      DYI = Y(I+1) - Y(I)
      B(I) = DYI/DXI - CON2*(A(I+1)-A(I))*DXI
    5 C(I) = Y(I+1) - CON2*A(I+1)*DXI*DXI - B(I)*X(I+1)
      RETURN
C
      ENTRY CINP (XI,YI,DY,D2Y)
      GO TO (66,10,11,11,13),ITPSZ
C
C     COMPUTE Y, DY/DX, D2Y/DX2 FROM CUBIC SPLINE COEFFICIENTS
   66 DO 6 I=1,NM
      IF (X(I) .LE. XI  .AND.  XI .LE. X(I+1)) GO TO 7
    6 CONTINUE
      WRITE (6,100) XI,X(1),X(N)
  100 FORMAT (7H0(CINP),5X,3HXI=,F13.5,17H  IS OUT OF RANGE/10X,5HX(1)=,
     1 F13.5,5X,5HX(N)=,F13.5)
C
    7 DXI = X(I+1) - X(I)
      A1 = (A(I+1)-A(I))/DXI/6.
      AIX = A(I)*X(I+1)
      AXI = A(I+1)*X(I)
      A2 = (AIX - AXI)/DXI/2.
      AIX = AIX*X(I+1)
      AXI = AXI*X(I)
      A3 = (AXI - AIX)/DXI/2. + B(I)
      AIX = AIX*X(I+1)
      AXI = AXI*X(I)
      A4 = (AIX - AXI)/DXI/6. + C(I)
C
      YI = F(XI)
      DY = DF(XI)
      D2Y = D2F(XI)
      RETURN
C
C     COMPUTE Y, DY/DX, D2Y/DX2 FROM INPUT POLYNOMIAL
   10 A1 = C1
      A2 = C2
      A3 = C3
      A4 = C4
C
      YI = F(XI)
      DY = DF(XI)
      D2Y = D2F(XI)
```

```
      RETURN
C
C     COMPUTE Y, DY/DX, D2Y/DX2 FROM INPUT SPECIAL FUNCTION
C       EXCEPTIONAL CASE AT X=0
   11 IF (XI .EQ. 0.) GO TO 12
      TERM = XI/FLM
      YI = G(TERM)
      DY = DG(YI)
      D2Y = D2G(DY,YI,XI)
      RETURN
C
   12 YI = 1.
C
C     FIT A CUBIC THROUGH THE POINTS (0.,YI),(.05,Y2'),(.05,Y2''), AND
C     (.10,Y3'') IN ORDER TO FIND YI' AND YI''
      TERM = .05/ELM
      Y2 = G(TERM)
      Y2P = DG(Y2)
      Y2PP = D2G(Y2P,Y2,.05)
      TERM = .10/ELM
      Y3 = G(TERM)
      Y3P = DG(Y3)
      Y3PP = D2G(Y3P,Y3,.10)
C
      DY = (.05*(Y3PP-Y2PP)/(.10-.05)/2. - Y2PP)*.05 + Y2P
      D2Y = Y2PP - .05*(Y3PP-Y2PP)/(.10-.05)
      RETURN
C
C     V=0 CASE - ASSIGNED AREA IS NOT REQUIRED
   13 YI = 1.
      DY = 0.
      D2Y = 0.
      RETURN
C
      END
```

```
CDERIV          DFFQ COMPATIBLE DERIVATIVE                                    DERIV001
      SUBROUTINE DERIV(Y,F,DFDY,J1,J2)                                        DERIV002
      DENOM(A,B,C)=(A-B)*(A-B)*(C-A)+(C-A)*(C-A)*(A-B)                        DERIV003
      DIMENSION  Y(2),F(2),DFDY(2)                                            DERIV004
    1 DO 10 J=J1,J2                                                           DERIV005
      DELA2=Y(J)-Y(J-1)                                                       DERIV006
      DELB2=Y(J+1)-Y(J)                                                       DERIV007
      DDD=1./DENOM(Y(J),Y(J-1),Y(J+1))                                        DERIV008
      DFDY(J)=(DELA2*DELA2*(F(J+1)-F(J))+DELB2*DELB2*(F(J)-F(J-1)))*DDD DERIV009
   10 CONTINUE                                                                DERIV010
   11 RETURN                                                                  DERIV011
      END                                                                     DERIV012
```

AD-A045 627 GENERAL ELECTRIC CO CINCINNATI OHIO AIRCRAFT ENGINE GROUP F/G 21/5
DEVELOPMENT OF EMISSIONS MEASUREMENT TECHNIQUES FOR AFTERBURNIN--ETC(U)
OCT 75 W C COLLEY, D R FERGUSON, M A SMITH F33615-73-C-2047
UNCLASSIFIED R75AEG459 AFAPL-TR-75-52-SUPPL-2 NL

3 OF 5
ADA045627



3 OF 5

ADA045627

1.0 1.1 1.25 1.4 1.6 1.8 2.0 2.2 2.5 2.8 3.2 3.6 4.0 4.5 5.0 5.6

MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

```
CDERVV      COMPUTE ALL DERIVATIVES WRT THE INDEPENDENT VARIABLE
      SUBROUTINE DERV
C
C     COMPUTE ALL DERIVATIVES WRT THE INDEPENDENT VARIABLE
C
      DOUBLE PRECISION DPS1,DPS2
C
      LOGICAL TCON,RHOCON
C
      REAL IVAR,MIXMW,M2
C
      COMMON/OPTS/VERSI,TIMEV,VERSA,AREAV,FLIM,TCON,RHOCON,IPRCOD
      COMMON/COND/DVAR,AREA,MDOT,P,IVAR,V,RHO,T,SIGMA(25),LS,LSP3,NEXT
      COMMON/SPEC/SNAM(2,30),MW(25),W(25),STOIC(25,30),OMEGA(25,30)
      COMMON/GHSC/GRT(25),HRT(25),SR(25),CPR(25),DCPR(25)
      COMMON/DERN/F(28),ALPHA(28),BETA(28,28)
      COMMON/NECC/RR,MIXMW,M2,GAMMA,TCPR,R
      COMMON/SABS/S1,AA,BB,S2,DA,D2A,DTERM,IRHO
C
      EQUIVALENCE (DV,F(1))
C
      D2ADT2(D2) = V*V*D2 + DV/V*DA
      D2ADX2(D2) = (D2/V - DV*DA)/V
C
      DPS1 = 0.D0
      DPS2 = 0.D0
      DO 1 I=1,3
      F(I) = 0.
    1 ALPHA(I) = 0.
C
      DENM = RHO
      IF (VERSI .NE. TIMEV) DENM = RHO*V
      DO 2 I=1,LS
      II = I + 3
C
C     DSIGMA/DIVAR
      F(II) = W(I)/DENM
C
C     S1 FOR AA
      DPS1 = DPS1 + F(II)
C     S2 FOR BB
    2 DPS2 = DPS2 + HRT(I)*F(II)
C
      S1 = MIXMW*DPS1
      S2 = MIXMW*DPS2
C
      GAM1 = GAMMA - 1.
C     BB FOR DERIVATIVES
      BB = GAM1/GAMMA*S2
C
```

```
C     AA FOR DERIVATIVES
       AA = S1 - BB
       ED=GAM1*(S2-S1)
       IF(RHOCON.AND.V.EQ.0..AND..NOT.TCON)IRHO=2
C
C     DAVAR/DIVAR
       IF (VERSI .EQ. TIMEV  .AND.  IPRCOD .LE. 2) DA = V*DA
       IF (VERSI .NE. TIMEV  .AND.  IPRCOD .GE. 3) DA = DA/V
C
       IF (VERSA .NE. AREAV) GO TO 4
C    ASSIGNED AREA EQUATIONS
       DTERM = DA/AREA - AA
       T1 = 1./(M2 - 1.)
       T2 = M2*T1
C
C     DV/DIVAR
       F(1) = V*T1*DTERM
C
C     DRHO/DIVAR
       IF (.NOT. RHOCON) F(2) = -RHO*(T2*DTERM + AA)
C
C     DT/DIVAR
       IF (.NOT. TCON) F(3) = -T*(GAM1*T2*DTERM + BB)
       IF(IRHO.EQ.2)F(3)=-T*ED
C
C     DAREA/DIVAR WRT IVAR
       IF (VERSI .EQ. TIMEV .AND. (IPRCOD .EQ. 1 .AND. V .NE. 0.)) D2A =
      * D2ADT2(D2A)
       IF (VERSI .NE. TIMEV  .AND.  IPRCOD .EQ. 3) D2A = D2ADX2(D2A)
C
       T3 = (D2A - DA*DA/AREA)/AREA
C
C     DSIGMA/DIVAR WRT IVAR
       DO 3 I=4,LSP3
     3 ALPHA(I) = 0.
C
C     DV/DIVAR WRT IVAR
       ALPHA(1) = V*T1*T3
C
C     DRHO/DIVAR WRT IVAR
       IF (.NOT. RHOCON) ALPHA(2) = -RHO*T2*T3
C
C     DT/DIVAR WRT IVAR
       IF (.NOT. TCON) ALPHA(3) = -T*GAM1*T2*T3
C
       GO TO 6
C    ASSIGNED PRESSURE EQUATIONS
     4 DTERM = DA/P
       T2 = -1./GAMMA
C
```

```
C     DV/DIVAR
       IF (V .NE. 0.) F(1) = -DA/(RHO*V)*1.01325E+06
C
C     DRHO/DIVAR
       IF (.NOT. RHOCON) F(2) = -RHO*(T2*DTERM + AA)
C
C     DT/DIVAR
       IF (.NOT. TCON) F(3) = -T*(GAM1*T2*DTERM + BB)
C
C     DP/DIVAR WRT IVAR
       IF (VERSI .EQ. TIMEV  .AND.  IPRCOD .EQ. 2) D2A = D2ADT2(D2A)
       IF (VERSI .NE. TIMEV  .AND.  IPRCOD .EQ. 4) D2A = D2ADX2(D2A)
C
       T3 = (D2A - DA*DA/P)/P
C
C     DSIGMA/DIVAR WRT IVAR
       DO 5 I=4,LSP3
     5 ALPHA(I) = 0.
C
C     DV/DIVAR WRT IVAR
       IF (V .NE. 0.) ALPHA(1) = -D2A/(RHO*V)*1.01325E+06
C
C     DRHO/DIVAR WRT IVAR
       IF (.NOT. RHOCON) ALPHA(2) = -RHO*T2*T3
C
C     DT/DIVAR WRT IVAR
       IF (.NOT. TCON) ALPHA(3) = -T*GAM1*T2*T3
C
     6 CALL PARD
C
       RETURN
       END
```

```
CDFEQ         GEN. ROUTINE-DIFFUSION EQUATION EQN--CENTERED/NON CENTERED DFEQ0001
      SUBROUTINE DFEQ(NDIFF,NBCL,NBCU,IERIND)                           DFEQ0002
      LOGICAL FIRST                                                     DFEQ0003
C*                                                                      DFEQ0004
      DENOM(AA,BB,CC)=(AA-BB)*(AA-BB)*(CC-AA)+(CC-AA)*(CC-AA)*(AA-BB)   DFEQ0005
C*                                                                      DFEQ0006
C*  PDEQ--  EPSLN*(DF/DX)=ALPHA*D(BETA*(DF/DPHI))/DPHI+GAMMA+DLTA*F     DFEQ0007
C*           +ETA*(DF/DPHI)                                             DFEQ0008
C*  FM1=SOLUTION VECTOR--PREVIOUS STATION DESTROYED BY CURRENT STATION  DFEQ0009
C*  DFM1=  DERIVATIVE VECTOR                                            DFEQ0010
C*  ALPHA,BETA,GAMMA,DLTA,EPSLN,ETA--COEFFICIENTS IN PDEQ               DFEQ0011
C*  DELM1,DELM=  NORMAL STEP SIZES AT PREVIOUS AND CURRENT STATIONS     DFEQ0012
C*  DELX=  AXIAL STEP SIZE                                              DFEQ0013
C*  B1,C1,D1,AN,BN,DN=  BOUNDARY COEFFICIENTS                           DFEQ0014
C                                                                       DFEQ0015
C*  USES NON CENTERED 3-POINT DIFFERENCES                               DFEQ0016
C*                                                                      DFEQ0017
C*                                                                      DFEQ0018
C*  STANDARD IMPLICIT FORM                                              DFEQ0019
C*                                                                      DFEQ0020
C*                                                                      DFEQ0021
C*  BRANCH FOR DISCONTINUOUS FUNCTION FORM REQUIRES INTERPOLATION       DFEQ0022
C*  AT EACH MESH POINT....                                              DFEQ0023
C*                                                                      DFEQ0024
C*                                                                      DFEQ0025
C*  NBC--NORMAL BC INDICATOR--L=LOWER,U=UPPER                           DFEQ0026
C*      =0 BACKWARD DIFFERENCE                                          DFEQ0027
C*      =1 USE TWO POINTS ABOVE OR BELOW BOUNDARY                       DFEQ0028
C*    EXTRA COEFFICIENTS STORED IN CARRY(1,1) AND/OR CARRY(3,NM)        DFEQ0029
C*      =2  USE B.C. WHICH SATISFY DIFFERENTIAL EQUATION                DFEQ0030
C*                                                                      DFEQ0031
C*                                                                      DFEQ0032
C*  NDIFF--INDICATES DIFFERENCE FORM FOR D(BETA*DF/DPHI)/DPHI           DFEQ0033
C*       =0  EXPAND 2-ND DERIVATIVE                                     DFEQ0034
C*       =1  USE FORM APPLICABLE FOR DISCONTINUOUS FUNCTIONS            DFEQ0035
C*                                                                      DFEQ0036
      COMMON /PARAM/                                                    DFEQ0037
     * ALPHA(200) , BETA(200)  , GAMMA(200) ,                           DFEQ0038
     * EPSLN(200) , DLTA(200)  ,                                        DFEQ0039
     * FM1(200)   , DFM1(200)  ,                                        DFEQ0040
     * PHI1(200)  , NM1        , PHI2(200) , NM        ,                DFEQ0041
     * DX  ,                                                            DFEQ0042
     * B1  , C1   , D1   ,                                              DFEQ0043
     * AN  , BN   , DN                                                  DFEQ0044
      COMMON /PARAM1/  ETA(200)                                         DFEQ0045
      COMMON /YDFXI/ NLC                                                DFEQ0046
      EQUIVALENCE (PHI(1),PHI2(1))                                      DFEQ0047
      DIMENSION CARRY(200,4)                                            DFEQ0048
      DIMENSION PHI(200)                                                DFEQ0049
C*                                                                      DFEQ0050
```

```
      DATA FIRST/T/,NUM/200/
C*
    1 IERIND=0
      IF(DX.EQ.0.) DX=1.
      NGOL=NBCL+1
      NGOU=NBCU+1
      NDF=NDIFF+1
      IF(PHI2(NM).LE.PHI1(NM1)) GO TO 10
C*
C*  ADD POINTS TO PREVIOUS STATION TO ACCOUNT FOR MESH SPREADING
C*
      DELM1=PHI1(NM1)-PHI1(NM1-1)
      DELTA=PHI2(NM)-PHI1(NM1)
      NEND=2+IFIX(DELTA/DELM1)+NM1
      J=NM1+1
    5 PHI1(J)=PHI1(J-1)+DELM1
      ALPHA(J)=ALPHA(NM1)
      BETA(J)=BETA(NM1)
      GAMMA(J)=GAMMA(NM1)
      EPSLN(J)=EPSLN(NM1)
      DLTA(J)=DLTA(NM1)
      ETA(J)=ETA(NM1)
    6 FM1(J)=FM1(NM1)
      J=J+1
      IF(J .LE. NEND) GO TO 5
      NM1=NEND
C*
   10 NMM1=NM-1
C*
C*  CONSTRUCT COEFFICIENT ARRAY FOR NODES 2 TO (NMM1)
C*  DELTA=S AT EACH STATION ARE SAME, BYPASS INTERPOLATION SECTION
C*
   12 J=1
   13 J=J+1
      DEL1=PHI1(J)-PHI1(J-1)
      DEL2=PHI1(J+1)-PHI1(J)
      DEL3=PHI2(J)-PHI2(J-1)
      DEL4=PHI2(J+1)-PHI2(J)
      Y=PHI2(J)
      DEL34=DEL3+DEL4
      DEL3SQ=DEL3*DEL3
      DEL4SQ=DEL4*DEL4
      ASSIGN 21 TO KGO
      IF(DEL1.EQ.DEL3 .AND. DEL2.EQ.DEL4) GO TO 19
      ASSIGN 37 TO KGO
C*
C*  LINEAR INTERPOLATION FOR ALP,GAM,FM2,EPS,BETA    ETC
C*
      NLC=1
   14 ALP=YOF(Y,PHI1,ALPHA,1,NM1)
```

```
      GAM=YDF(Y,PHI1,GAMMA,1,NM1)                                       DFEQ0101
      FM2=YDF(Y,PHI1,FM1    ,1,NM1)                                     DFEQ0102
      EPS=YDF(Y,PHI1,EPSLN,1,NM1)                                       DFEQ0103
      DLTA1=YDF(Y,PHI1,DLTA,1,NM1)                                      DFEQ0104
      ETA1=YDF(Y,PHI1,ETA,1,NM1)                                        DFEQ0105
      GO TO (15,16),NDF                                                 DFEQ0106
   15 W=Y+DEL4                                                          DFEQ0107
      Z=Y-DEL3                                                          DFEQ0108
      BETL=YDF(Z,PHI1,BETA,1,NM1)                                       DFEQ0109
      BET =YDF(Y,PHI1,BETA,1,NM1)                                       DFEQ0110
      BETU=YDF(W,PHI1,BETA,1,NM1)                                       DFEQ0111
      GO TO 30                                                          DFEQ0112
C*                                                                      DFEQ0113
C*  DISCONTINUOUS FUNCTION(BETA*DF/DPHI) BRANCH                         DFEQ0114
C*                                                                      DFEQ0115
   19 GO TO (20,16),NDF                                                 DFEQ0116
   16 W=Y+.5*DEL4                                                       DFEQ0117
      Z=Y-.5*DEL3                                                       DFEQ0118
      NLC=1                                                             DFEQ0119
      BMHALF=YDF(Z,PHI1,BETA,1,NM1)                                     DFEQ0120
      EMHALF=YDF(Z,PHI1,ETA,1,NM1)                                      DFEQ0121
      BPHALF=YDF(W,PHI1,BETA,1,NM1)                                     DFEQ0122
      EPHALF=YDF(W,PHI1,ETA,1,NM1)                                      DFEQ0123
      IF(J.NE. 2) GO TO 161                                             DFEQ0124
      BSV=BMHALF                                                        DFEQ0125
      ESV=EMHALF                                                        DFEQ0126
  161 IF(J.NE. NMM1) GO TO 20                                           DFEQ0127
      BSVU=BPHALF                                                       DFEQ0128
      ESVU=EPHALF                                                       DFEQ0129
   20 GO TO KGO , (21,37)                                               DFEQ0130
   21 ALP=ALPHA(J)                                                      DFEQ0131
      GAM=GAMMA(J)                                                      DFEQ0132
      EPS=EPSLN(J)                                                      DFEQ0133
      DLTA1=DLTA(J)                                                     DFEQ0134
      ETA1=ETA(J)                                                       DFEQ0135
      FM2=FM1(J)                                                        DFEQ0136
      IF(NDF.EQ.2) GO TO 37                                             DFEQ0137
C*                                                                      DFEQ0138
C*  EXPANDED 2-ND DERIVATIVE BRANCH----                                 DFEQ0139
C*  D(BETA*DF/DPHI)/DPHI=DBETA/DPHI*DF/DPHI+BETA*D2F/DPHI2              DFEQ0140
C*                                                                      DFEQ0141
      BET=BETA(J)                                                       DFEQ0142
      BETU=BETA(J+1)                                                    DFEQ0143
      BETL=BETA(J-1)                                                    DFEQ0144
   30 DD=1./DENOM(PHI2(J),PHI2(J-1),PHI2(J+1))                          DFEQ0145
      DBETA=(DEL3SQ*(BETU-BET)+DEL4SQ*(BET-BETL))*DD                    DFEQ0146
   32 TSV=ALP*DX*DD                                                     DFEQ0147
      TL=2.*TSV*BET                                                     DFEQ0148
      TM=TSV*DBETA                                                      DFEQ0149
      TN=GAM*DX                                                         DFEQ0150
```

```
      TK=ETA1*DX*DD
      TMK=TK+TM
C*
   35 CARRY(J,1)=TMK*DEL4SQ-TL*DEL4
      CARRY(J,2)=EPS+TL*DEL34+TMK*(DEL3SQ-DEL4SQ)
     * -DLTA1*DX
      CARRY(J,3)=-TMK*DEL3SQ-TL*DEL3
      CARRY(J,4)=EPS*FM2+TN
      GO TO 39
C*
   37 TSV=2.*ALP*DX/DEL34
      TL=TSV*BPHALF/DEL4
      TM=TSV*BMHALF/DEL3
      TN=GAM*DX
      TKM=.5*EMHALF/DEL3
      TKP=.5*EPHALF/DEL4
      CARRY(J,1)=-TM+TKM
      CARRY(J,2)=EPS+TM+TL-DLTA1*DX+TKP-TKM
      CARRY(J,3)=-TL-TKP
      CARRY(J,4)=EPS*FM2+TN
   39 IF(J.LT.NMM1) GO TO 13
C*
C*  STORE UPPER AND LOWER B.C. IN CARRY
C*
      DELT1=PHI(2)-PHI(1)
      GO TO (40,46,400),NGOL
C*
   40 CARRY(1,1)=0.
      CARRY(1,2)=B1-C1/DELT1
      CARRY(1,3)=C1/DELT1
      CARRY(1,4)=D1
      GO TO 47
C*
   46 CARRY(1,1)=-C1*PHI(2)/(PHI(3)*(PHI(3)-PHI(2)))
      CARRY(1,2)=B1-(PHI(2)+PHI(3))/(PHI(2)*PHI(3))*C1
      CARRY(1,3)=-CARRY(1,1)*PHI(3)**2/PHI(2)**2
      CARRY(1,4)=D1
      GO TO 47
C*
  400 DELT2=1./DELT1
      DELT1S=DELT2**2
      CARRY(1,1)=0.
      TERM=2.*C1*ALPHA(1)*BSV*DELT1S
      TERM2=2.*D1*ALPHA(1)*DELT2
      CARRY(1,2)=2.*B1*ALPHA(1)*DELT2-TERM-C1*EPSLN(1)/DX
     * +C1*DLTA(1)-C1*FSV*DELT1
      CARRY(1,3)=TERM+C1*ESV*DELT1
      CARRY(1,4)=-C1*(GAMMA(1)+FM1(1)*EPSLN(1)/DX)+TERM2
   47 DELTN=PHI(NM)-PHI(NMM1)
      GO TO (48,49,410),NGOU
```

```
C*
   48 CARRY(NM,1)=-BN/DELTN
      CARRY(NM,2)=AN+BN/DELTN
      CARRY(NM,3)=0.
      CARRY(NM,4)=DN
      GO TO 50
C*
   49 PHIM1=PHI(NMM1)*PHI(NMM1)
      PHIM2=PHI(NM-2)*PHI(NM-2)
      ZTA=(2.*PHI(NM)*PHI(NMM1)-PHIM1)/
     * (PHIM2+2.*PHI(NM)*(PHI(NMM1)-PHI(NM-2))-PHIM1)
      DEN=1./((1.-ZTA)*PHI(NMM1)+ZTA*PHI(NM-2))
      CARRY(NM,1)=(1.-ZTA)*BN*DEN
      CARRY(NM,2)=AN-BN*DEN
     * -BN*DLTA(NM)
      CARRY(NM,3)=ZTA*BN*DEN
      CARRY(NM,4)=DN
      GO TO 50
C*
  410 DELT1=1./(PHI(NM)-PHI(NMM1))
      DELT1S=DELT1**2
      TERM=-2.*BN*ALPHA(NM)*BSVU*DELT1S
      TERM2=2.*DN*ALPHA(NM)*DELT1
      CARRY(NM,1)=TERM+BN*ESVU*DELT1
      CARRY(NM,2)=2.*AN*ALPHA(NM)*DELT1+BN*EPSLN(NM)/DX-TERM
     * -BN*ESVU*DELT1
      CARRY(NM,3)=0.
      CARRY(NM,4)=BN*(GAMMA(NM)+EPSLN(NM)*FM1(NM)/DX)+TERM2
C*
C*  CALL TDSEQ FOR SOLUTION TO SIMULTANEOUS EQUATIONS.
C*
C*
C*  FIRST CALL SETS UP ADDRESSES FOR TDSEQ---BYPASS ON SUBSEQUENT
C*
   50 IF(.NOT. FIRST) GO TO 51
      CALL TDSEQ(CARRY,NM,NUM,ERR)
      FIRST=.FALSE.
      GO TO 52
   51 CALL TDSEQ1
   52 IF(ERR.GT. 0.) IERIND=1
C*
C*  STORE SOLUTION IN FM1 AND COMPUTE DF/DPHI
C*
      FM1(1)=CARRY(1,1)
      FM1(NM)=CARRY(NM,1)
      DELT1=PHI(2)-PHI(1)
      DELTN=PHI(NM)-PHI(NMM1)
      IF(B1 .EQ.0. .AND. D1.EQ.0.) DFM1(1)=0.
      IF(AN .EQ. 0. .AND. DN.EQ.0.) DFM1(NM)=0.
      J=2
```

```
   61 FM1(J)=CARRY(J,1)
      FM1(J+1)=CARRY(J+1,1)
      DELA2=PHI(J)-PHI(J-1)
      DELB2=PHI(J+1)-PHI(J)
      DDD=1./DENOM(PHI2(J),PHI2(J-1),PHI2(J+1))
      DFM1(J)=(DELA2*DELA2*(FM1(J+1)-FM1(J))+DELB2*DELB2
     * *(FM1(J)-FM1(J-1)))*DDD
      J=J+1
      IF(J.LE.NMM1) GO TO 61
      DFM1(1)=(FM1(2)-FM1(1))/DELT1
      DFM1(NM)=(FM1(NM)-FM1(NMM1))/DELTN
C*
C*
C*
      NM1=NM
  100 RETURN
      END
```

```
CEQGASH      THERMO PRPTYS OF EQLBRM COMBN GAS - H INDEP.                 EQGASH01
      SUBROUTINE EQGASH(FAR,WAR,HC,T,P,FIXCO,FIXNO,Z,                     EQGASH02
     1H,MWT,S,SPV,A,CP)                                                   EQGASH03
      REAL MWT,Z(11),QV(8)                                                EQGASH04
      LOGICAL FIXCO,FIXNO                                                 EQGASH05
      COMMON/COIREM/HTOL,H0,DHDT,CTRMAX                                   EQGASH06
      HTOL=0.01                                                           EQGASH07
      H0=H                                                                EQGASH08
      QV(1)=0.0                                                           EQGASH09
   10 IF(T.LT.189.0)T=189.0                                               EQGASH10
      IF(T.GT.9000.0)T=9000.0                                             EQGASH11
      CALL EQGAST(FAR,WAR,HC,T,P,FIXCO,FIXNO,Z,H1,MWT,S,SPV,A,CP)         EQGASH12
      IF((T.EQ.189.0).AND.(H1.GT.H0))GO TO 20                             EQGASH13
      IF(QV(1).EQ.25.0)GO TO 20                                           EQGASH14
      DHDT=CP                                                             EQGASH15
      CALL QIREM(T,H1,-5000.0,QV)                                         EQGASH16
      IF(QV(1).NE.0.0)GO TO 10                                            EQGASH17
      RETURN                                                              EQGASH18
C                                                                         EQGASH19
   20 WRITE(6,30)T,H1,H0,CP,FAR                                           EQGASH20
   30 FORMAT(27H0EQGASH - NOT CONVERGED. T=,F7.2,4H  H=,F8.2,             EQGASH21
     15H  H0=,F8.2,5H  CP=,F8.5,6H  FAR=,F9.6)                            EQGASH22
      RETURN                                                              EQGASH23
      END                                                                 EQGASH24
```

```
CEQGAST      THERMO PRPTYS OF EQLBRM COMBN GAS - T INDEP.
      SUBROUTINE EQGAST(FAR,WAR,HC,T,P,FIXCO,FIXNO,Z,
     1H,MWT,S,SPV,A,CP)
      REAL MWT,Z(11),JJ
      LOGICAL FIXCO,FIXNO
      COMMON/GHSC/FF(25),HH(25),SS(25),CCP(25),DCPDT(25)
C
C   ORDER OF SPECIES = H,O,H2,O2,OH,H2O,CO,CO2,N2,A,NO
C   IF FIXCO, Z(CO) AND Z(CO2) MUST BE PRESET
C   IF FIXNO, Z(NO) AND Z(N2) MUST BE PRESET
C
C   GET EQUILIBRIUM COMPOSITION (MOLES/LB MIX)
C
      IF(FAR.LT.0.0)FAR=0.0
      IF(WAR.LE.0.0)WAR=1E-8
      CALL HCEQ2(FAR,WAR,HC,T,P,FIXCO,FIXNO,Z)
C
C   ENTRY FOR DETERMINATION OF THERMO PROPERTIES OF
C   PREDETERMINED MIXTURES.
C
      ENTRY FRGAST(T,P,Z,H,MWT,S,SPV,A,CP)
      CALL THRM(T/1.8,1.0)
C
C   SPECIFIC MOLARITY (1/MWT)
C
   10 SPM=0.0
      DO 20 I=1,11
   20 SPM=SPM+Z(I)
C
C   THERMOSTATIC PROPERTIES
C
      MWT=1.0/SPM
      DATA RO,GO,JJ/1.98596,32.1740,778.2/
      CP=0.0
      H=0.0
      S=0.0
      DO 30 I=1,11
      CP=CP+Z(I)*CCP(I)
      H=H+Z(I)*HH(I)
   30 S=S+Z(I)*(HH(I)-FF(I)-ALOG(AMAX1(Z(I),1E-38)))
      CP=CP*RO
      H=H*RO*T
      S=(S-SPM*ALOG(MWT*P/14.696))*RO
      PSF=144.0*P
      SPV=RO*JJ*T*SPM/PSF
      GAM=CP*MWT/RO
      GAM=GAM/(GAM-1.0)
      A=SQRT(PSF*SPV*GAM*GO)
      RETURN
      END
```

```
CEQKIN        EQUILIBRIUM TERMS FOR KINETICS CALCULATION                    EQKIN001
      SUBROUTINE EQKIN(P,H,FIXCO)                                           EQKIN002
C                                                                           EQKIN003
      COMMON /PSEQX / HOC,HUM,CO2AIR,MAIR,FS,FUELMW                         EQKIN004
      REAL                            MAIR                                  EQKIN005
      COMMON /GHSC  / FF(25),HZ(25),SR(25),CPZ(25),DCPR(25)                 EQKIN006
      COMMON /CEQKIN/ BK,AIK,DELHNO,DEN,XMWEQ,XNO,XCO                       EQKIN007
      COMMON /KININS/ XMWUN,XMWC,TCONST,CONER                               EQKIN008
      LOGICAL  TCONST                                                       EQKIN009
      COMMON /PSEQ  / FOA,BETS,TP,X(16),DHQDMW,TEQ,BEQ,XMWT,HNEQ            EQKIN010
      EQUIVALENCE (X1,X(1)),(X2,X(2)),(X3,X(3)),(X4,X(4)),                  EQKIN011
     *            (X5,X(5))                                                 EQKIN012
      COMMON /PROPR / PR,HHR,TR,FCR,RHOR,RR,WMWTR                           EQKIN013
      COMMON /CBITS / BITS,BLANK                                            EQKIN014
      COMMON /CKINET/ DUMK(89),RK1,RK2,RK3,RK4,RK5,TK,ENTRY1                EQKIN015
      LOGICAL          ENTRY1                                               EQKIN016
      COMMON /CCOCO2/ CORATE,XCOI                                           EQKIN017
      LOGICAL          CORATE                                               EQKIN018
      COMMON /CPRINT/ PDUM(20)                                              EQKIN019
      COMMON /SNMW  / ALSP(150), WMT(75)                                    EQKIN020
      COMMON/DOIREM/KINFIL                                                  EQKIN021
      LOGICAL FIXCO                                                         EQKIN022
      DIMENSION QV(8)                                                       EQKIN023
      DATA R0/1.98596/                                                      EQKIN024
      DATA C2/.01603286/                                                    EQKIN025
C                                                                           EQKIN026
C     GET EQUILIBRIUM BETA,TEMP,ETC.                                        EQKIN027
C                                                                           EQKIN028
      X(7)  = XCO                                                           EQKIN029
      X(11) = XNO                                                           EQKIN030
      HH    = H                                                             EQKIN031
      TEQ1  = TP                                                            EQKIN032
      XJP   = 0.1*TEQ1                                                      EQKIN033
      IF(PDUM(20).LE.0.0)GO TO 10                                           EQKIN034
      KINFIL=PDUM(20)+0.1                                                   EQKIN035
      REWIND KINFIL                                                         EQKIN036
      WRITE(KINFIL,1001)P,HH,FOA,HUM,HOC,XCO,XNO                            EQKIN037
 1001 FORMAT(45H **EQKIN**       TEMPERATURE ITERATION HISTORY,87X/         EQKIN038
     '3H P=,F8.4,6H    H=,F10.4,8H     FOA=,F9.6,8H     HUM=,F9.6,          EQKIN039
     '8H    H/C=,F6.3,8H    XCO=,E13.6,8H     XNO=,E13.6,15X/132X)          EQKIN040
   10 CALL HCEQ2(FOA,HUM,HOC,TEQ1,P,FIXCO,.TRUE., X)                        EQKIN041
      XMWEQ = 0.                                                            EQKIN042
      HH1   = 0.                                                            EQKIN043
      DO 12 K=1,11                                                          EQKIN044
      XMWEQ = XMWEQ+X(K)*WMT(K)                                             EQKIN045
   12 HH1   = HH1+X(K)*HZ(K)                                                EQKIN046
      HH1   = R0*TEQ1*HH1/XMWEQ                                             EQKIN047
      ERR   = HH-HH1                                                        EQKIN048
      IF(PDUM(20).GT.0.0)WRITE(KINFIL,1002)TEQ1,XMWEQ,HH1,(X(I),I=1,12)     EQKIN049
 1002 FORMAT(3H T=,F10.4,5H  MW=,F10.6,5H  H1=,F10.4,27X,2HX=,              EQKIN050
```

```
     14E15.8/12X,8E15.8)
      CALL QIREM(TEQ1,ERR,XJP,QV)
      IF( QV(1).NE.0. ) GO TO 10
      KINFIL=40
      TEQ    = TEQ1
      GAR    = 1.+FOA+HUM
      SUMH   = (FOA*HQC/FUELMW+HUM*2./18.016)/GAR
      SUMC   = (FOA/FUELMW+CO2AIR/MAIR)/GAR
      SUMO   = (HUM/18.016+2.*(.209495+CO2AIR)/MAIR)/GAR
      SUMN   = 2.*.780881/(MAIR*GAR)
      SUMA   = (.009624-CO2AIR)/(MAIR*GAR)
      ZZX    = .5*(CO2AIR+MAIR/18.016*HUM)*SUMN/.780881
      SPMU   = .5*(SUMO+SUMC+SUMH+SUMN-ZZX)+SUMA
      SPMC   = .5*(.5*SUMH+SUMN+SUMO)+SUMA
      IF( FOA.GT.FS ) SPMC=.5*(SUMH+SUMN)+SUMC+SUMA
      XMWUN = 1./SPMU
      XMWC   = 1./SPMC
      DEN    = 1./(SPMC-SPMU)
      BEQ    = (1./XMWEQ-1./XMWUN)*DEN
C
C     CALCULATE TERMS FOR KINETICS
C
      RHOS   = 144.*XMWEQ*P/(1545.32*TEQ)
      RHOCGS= C2*RHOS
      TK     = TEQ/1.8
      TEMP   = TP
      TP     = TEQ
      CALL RATCON( P )
   60 TP     = TEMP
      RATE   = -RHOCGS**2/XMWEQ**3*(RK1*X1*X5+RK2*X1**2+RK3*X1*X2+
     *                                  RK4*X2**2+RK5*X1*X4)
      BK     = 0.
      IF( BEQ.GE.1. ) GO TO 100
      B      = RATE/((1./XMWEQ-1./XMWC)**2*DEN)
      BK     = B*(1.-BEQ)**2
C
  100 RETURN
      END
```

```
CERROR1     QIREM QUITS                                              ERROR101
      SUBROUTINE ERROR1                                              ERROR102
      LOGICAL NODATA                                                 ERROR103
      DIMENSION A(22)                                                ERROR104
      COMMON/CQIREM/YTOL,Y0,DYDX,TUMANY/DQIREM/NFILE                 ERROR105
      CALL FLGEOF(NFILE,NODATA)                                      ERROR106
      END FILE NFILE                                                 ERROR107
      REWIND NFILE                                                   ERROR108
      WRITE(6,100)YTOL,Y0,DYDX,TUMANY                                ERROR109
  100 FORMAT(6H0YTOL=,E16.8,5H  Y0=,E16.8,7H  DYDX=,E16.8,9H  TUMANY=, ERROR110
     'E16.8)                                                         ERROR111
      DO 10 LINES=1,3000                                             ERROR112
      READ(NFILE,110)A                                               ERROR113
      IF(NODATA)GO TO 20                                             ERROR114
   10 WRITE(6,110)A                                                  ERROR115
  110 FORMAT(22A6)                                                   ERROR116
   20 CALL ERROR                                                     ERROR117
      RETURN                                                         ERROR118
      END                                                            ERROR119
```

```
CERRORC     RELATIVE ERROR IN INTEGRATION STEP
      FUNCTION ERRORC(Y,RK,E,JC,H)
C
C     COMPUTE THE RELATIVE ERROR IN AN INTEGRATION STEP OF SIZE H
C
C     DETERMINE THE CONTROLLING VARIABLE
C
      DOUBLE PRECISION DNP1,DN,DNM1,C,ABSY
C
      DIMENSION Y(28),RK(28),E(28)
C
      COMMON/COND/DUM1(33),LS,LSP3,NEXT
      COMMON/SINT/HMIN,HNM1,HN,HNP1,HMAX,NH,AVH,EMAX,ERRN,JCV,KOUNT,ERRP
      COMMON/PQRF/PK(28),QK(28),DUM2(56)
      COMMON/SKIP/NEGL(25),I1,I2,IT
C
      C = (H + HN)/(HN + HNM1)
      FAC1 = ABS(H*H/(2.*H + HN)*(2.*H + HN)/(H + HN + HNM1))
C
      ERRORC= -1.
      DO 4 I=1,LSP3
      E(I) = 0.
      IF (ABS(RK(I)) .EQ. 0.) GO TO 4
      ABSY = ABS(Y(I))
      IF (ABSY .EQ. 0.) GO TO 4
      DNP1 = RK(I)/H
      DN = QK(I)/HN
      DNM1 = PK(I)/HNM1
      FAC2 = DABS((DNP1 - DN) - C*(DN - DNM1))
C     CHECK FOR CATASTROPHIC SUBTRACTION
      IF (FAC2 .LE. ABS(DNP1)*1.0E-04) GO TO 5
      E(I) = FAC1*(FAC2/ABSY)
    5 IF (I .LE. 3  .OR.  IT .EQ. 0) GO TO 3
      II = I - 3
C     SKIP NEGLECTED SPECIES
      DO 2 J=1,IT
      IF (NEGL(J) .EQ. II) GO TO 4
    2 CONTINUE
    3 IF( E(I) .LE. ERRORC ) GO TO 4
      ERRORC= E(I)
      JC = I
    4 CONTINUE
C
      RETURN
      END
```

```
*FILL
      SUBROUTINE FILL(X,Y,NA,NB)
CFILL
C     LINEAR INTERPOLATION TO FIL VACANCIES IN INPUT LISTS
      COMMON /CBITS/BITS
      DIMENSION X(10),Y(10)
C     FIND IA,IB - VACANT REGION
      IA=NA+1
      IF(Y(IA-1).EQ.BITS) GO TO 99
    3 DO 4 I=IA,NB
      IF(Y(I).NE.BITS) GO TO 5
    4 CONTINUE
      IB=NB
      GO TO 7
    5 IB=I-1
      IF(I.EQ.IA) GO TO 12
C     FILL VACANCIES
      IF(Y(IB+1).NE.Y(IA-1)) GO TO 9
C     ALL VALUES THE SAME
    7 DO 8 II=IA,IB
    8 Y(II)=Y(IA-1)
      GO TO 12
C     INTERPOLATE
    9 DX = X(IB+1) - X(IA-1)
      DO 11 II=IA,IB
   11 Y(II) = (Y(IB+1)*(X(II)-X(IA-1)) + Y(IA-1)*(X(IB+1)-X(II)))/DX
C     GO BACK AND SEARCH FOR MORE REGIONS
   12 IA = IB+2
      IF(I.LT.NB) GO TO 3
   99 RETURN
      END
```

```
CFUEL        THERMO PROPERTIES OF RAW FUEL VAPOR
C
      SUBROUTINE FUEL(N,TR,H,CP)
C
      REAL N,C(7,2)
C
C    N IS H/C RATIO OF FUEL
C     C ARE NASA COEFFICIENTS FOR C2H4
C
      DATA TLOW,TMID,THI/300.0,1000.0,5000.0/,
     'C/3.4552152,1.1491803E-2,-4.3651750E-6,7.6155095E-10,
     '-5.0123200E-14,6.7694946E3,2.6987959,
     '1.4256821,1.1383140E-2,7.9890006E-6,-1.6253679E-8,
     '6.7491256E-12,7.6292582E3,1.4621819E1/
C
      PNOM(T)=A1+T*(A2+T*(A3+T*(A4+T*A5)))
      TK=TR/1.8
C
      IF(TK.GE.0.35*TLOW.AND.TK.LE.THI)GO TO 20
      WRITE(6,999)TR
      IF(TK.GT.0.0.AND.TK.LE.1.20*THI)GO TO 10
      CALL ERROR
   10 WRITE(6,998)
C
  999 FORMAT(10H0FUEL TEMP,F9.1,22H DEGR IS OUT OF RANGE.)
  998 FORMAT(30H EXTRAPOLATED VALUES RETURNED.)
C
C    SELECT TEMPERATURE RANGE
   20 K=2
      IF(TK.GT.THID)K=1
C
C    COMPUTE ENTHALPY OF ETHYLENE
C
      A1=C(1,K)+C(6,K)/TK
      A2=C(2,K)/2.0
      A3=C(3,K)/3.0
      A4=C(4,K)/4.0
      A5=C(5,K)/5.0
      HRT=PNOM(TK)
      H=HRT*1.98596*TR/28.054
C
C    COMPUTE SPECIFIC HEAT
C
      A1=C(1,K)
      A2=C(2,K)
      A3=C(3,K)
      A4=C(4,K)
      A5=C(5,K)
      CP=PNOM(TK)*1.98596/28.054
C
```

```
C   GE STANDARD GUESS AT JET FUEL HEATING VALUE                          FUEL0051
C                                                                        FUEL0052
      HV=(184686.04+37977.7*N)/(11.91468+N)                              FUEL0053
C                                                                        FUEL0054
C   HEAT OF VAPORIZATION AT 77F IS ABOUT 155 B/LB.                       FUEL0055
C                                                                        FUEL0056
      H=H-20275.0+HV+155.0                                               FUEL0057
      RETURN                                                             FUEL0058
      END                                                                FUEL0059
```

```
CGAMCP          COMPUTE GAMMA, CP----   (P.H. HECK)
      SUBROUTINE GAMCP( TT,GAM,CP,RGAS,J1,J2)
      DIMENSION TT(1),GAM(1),CP(1)
C*
C*  VALID TEMPERATURE RANGE=    T.LT.3600
C*
    1 ROJ=RGAS/778.
      DO 10 L=J1,J2
      T=TT(L)
      IF(T.LE.800.) GO TO 50
      IF(T.GE.3600.) GO TO 40
C*
      GAM(L)=2.23708/T**.070271
      GO TO 10
C*
   50 GAM(L)=1.4
      GO TO  10
C*
   40 GAM(L)=1.294
C*
   10 CP(L)=GAM(L)*ROJ/(GAM(L)-1.)
      RETURN
      END
```

```
CGAMH          FUNCTION  GAMH  (T)--P.H. HECK                GAMH0001
      FUNCTION GAMH(T)                                        GAMH0002
      IF(T.LE. 800.) GO TO 10                                 GAMH0003
      IF(T.GE. 3600.) GO TO 12                                GAMH0004
C*                                                            GAMH0005
      GAMH=2.23708/T**.070271                                 GAMH0006
      GO TO 15                                                GAMH0007
   10 GAMH=1.4                                                GAMH0008
      GO TO 15                                                GAMH0009
   12 GAMH=1.254                                              GAMH0010
   15 RETURN                                                  GAMH0011
      END                                                     GAMH0012
```

```
CGCKP         MAIN DRIVER ROUTINE FOR GCKP-I CALC.
      SUBROUTINE GCKP( KK )
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
      LOGICAL         ERR,ERRMAJ,INERR,PRERR
      COMMON /INDATA/ N,HF,WAR,T2,BETA,T25,FAR5,EINO2C,P0,
     *                RCO(11),RCO2(11),RHC(11),RNOX(11),
     *                PT(11),PS(11),BLOC(11),QCO(11)
      REAL            N
      COMMON /JETDAT/ NPTS,RAD(12),TS(12),U(12),SPV(12),MWT(12),
     *                CP(12),FUU (12),SPALDG(12),TKE(12),OTHER1(36)
      REAL            MWT
      COMMON /GASCMP/ RICH(12,2),FUEL(2,12,2),ENTH(2,12,2),
     *                CONC(16,2,12,2),HCINCP(12,2),OTHER2(192)
      COMMON /GASTHW/ TG(2,12,2),MWTG(2,12,2),TAU(12,2),CPG(2,12,2)
      REAL            MWTG
      COMMON /STCTRL/ LSTA,FINAL,CHEMK,FIRSTM,FIRSTC,XC,DXC,
     *                NIST,POUT1,DUMST1(8)
      LOGICAL         FINAL,FIRSTM,FIRSTC
      LOGICAL         POUT1
      INTEGER         CHEMK
      COMMON /CPRINT/ PDUM(20)
      COMMON /COND  / DVAR,AREA,MDOT,P,IVAR,V,RHO,T,C(25),LS,DUM2(2)
      COMMON /PRIN  / PRINT(50),NPRNTS,END,EVSTEP
      REAL            IVAR
C
      POUT1 = .FALSE.
      K     = KK
C
C     SET GCKP COMMONS
C
      P     = P0*144.
      T     = TG(1,K,2)
      IF( T.LE.1500. ) GO TO 201
      DVAR  = 0.
      IVAR  = TAU(K,1)
      NPRNTS = 1
      PRINT(1)= TAU(K,2)
      XMW   = MWTG(1,K,2)
      CALL EMPYC(1,XMW,CONC(1,1,K,2),C,16)
      IF( FINAL .AND. LSTA.EQ.1 .AND.K.EQ.1 ) POUT1 =.TRUE.
      IF( FINAL ) WRITE (6,50) K
      IF( PDUM(11).NE.0. ) WRITE (6,50) K
   50 FORMAT(1H1,/////////////////////////////47X,27H***   S T R E A M   T
     *U B E,2X,I2,3X,3H***)
      CALL GCKP1( $199 )
      FIRSTC= .FALSE.
      CALL MOVE(1,C,CONC(1,1,K,1),16,1)
      TG(1,K,1)= T*1.8
      GO TO 200
  199 ERR   = .TRUE.
```

```
200 RETURN                                                  GCKP0051
201 TG(1,K,1)= TG(1,K,2)                                    GCKP0052
    CALL MOVE(1,CONC(1,1,K,2),CONC(1,1,K,1),16,1)           GCKP0053
    GO TO 200                                               GCKP0054
    END                                                     GCKP0055
```

```
CGCKP1       GENERAL CHEMICAL KINETICS PROGRAM
      SUBROUTINE GCKP1(*)
C          OBTAINED FROM DA BITTKER OF NASA 11/29/71
C
      LOGICAL NEXT,EVSTEP
C
      REAL IVAR
C
      COMMON/COND/DUM1(4),IVAR,Y(28),DUM2(2),NEXT
      COMMON/SINT/HMIN,HINT,HN,HNP1,HMAX,NH,AVH,EMAX,ERRN,JCV,KOUNT,ERRP
      COMMON/PORF/PK(28),QK(28),RK(28),F(28)
      COMMON/PRIN/PRINT(50),NPRNTS,END,EVSTEP
      COMMON /STCTRL/ ISTA,FINAL,CHEMK,FIRSTM,FIRSTC,XC,DXC,
     *                NIST,POUT1,DUMST1(8)
      LOGICAL         FIRSTC,FINAL
      LOGICAL         POUT1
      COMMON /CPRINT/ PDUM(20)
C
C     READ AND CONVERT INPUT, PERFORM PRE-KINETIC CALCULATIONS
      CALL KINP
    1 IF (NEXT) GO TO 1000
C
C     PRINT REACTIONS, ASSIGNED VARIABLE PROFILE, INTEGRATION CONTROLS
C
      IF( FINAL .AND. POUT1  ) CALL OUT1
C
C     COMPUTE (NON-INPUT) INITIAL CONDITIONS
      CALL PRED1
C
C     PRINT ALL INITIAL CONDITIONS
      IF( FINAL ) CALL OUT2
      IF( PDUM(11).NE.0. ) CALL OUT2
      IF (NEXT) GO TO 1000
C
C     INITIAL INTEGRATION STEPS
      CALL INTI
      NH = 2
      AVH = HINT
      IF (NEXT) GO TO 1000
C
      IF (.NOT. EVSTEP) GO TO 3
C
C ** INTEGRATION  -  PRINT RESULTS AFTER EVERY STEP
    2 NH = 0
      PREV = IVAR
      CALL INTG
      NH = NH + 1
      AVH = HN
      IF (NH .NE. 1) AVH = (IVAR - PREV)/FLOAT(NH)
      IF( FINAL ) CALL OUT3
```

```
      IF (NEXT) GO TO 1000
      HTOP = END - IVAR
      IF (HTOP .LE. 0.) GO TO 100
      IF ((HTOP-HNP1) .LT. 0.) HNP1 = HTOP
      GO TO 2
C
C     LOCATE FIRST PRINT STATION
    3 DO 4 I=1,NPRNTS
      IF (PRINT(I) .GT. IVAR) GO TO 5
    4 CONTINUE
    5 NS = I
C
C ** INTEGRATION  -  PRINT RESULTS AT PRINT STATIONS
      DO 10 I=NS,NPRNTS
      NH = 0
      PREV = IVAR
C
C     SET NEXT PRINT STATION
      PRNT = PRINT(I)
C     INTEGRATE TO PRINT STATION
    6 HTOP = PRNT - IVAR
      IF (HTOP .LE. HNP1) GO TO 7
      CALL INTG
      NH = NH + 1
      IF (NEXT) GO TO 8
      GO TO 6
C
C     SPECIAL STEP TO PRINT STATION
    7 CALL CASG (HN,QK,HTOP,RK)
      ERRN = ERRORC(Y,RK,E,JCV,HTOP)
      CALL PRED
      NH = NH + 1
    8 AVH = (IVAR - PREV)/FLOAT(NH)
      IF( FINAL ) CALL OUT3
      IF( PDUM(11).NE.0. ) CALL OUT3
      IF (NEXT) GO TO 1000
      CALL INTC
C
   10 CONTINUE
C
  100 RETURN
C
 1000 WRITE (6,1001)
 1001 FORMAT (7H0(GCKP),5X,46HA FATAL ERROR HAS OCCURRED  -  CASE TERMIN
     *ATED)
      RETURN 1
C
      END
```

```
CGMOD       MODIFY INITIAL G PROFILES TO FORCE RAPID
C           HOMOGENIZATION OF LOW YRICH'S
C
      SUBROUTINE GMODFY
      COMMON /CBITS / BITS,BLANK
      REAL MJET,MUREF,N
      DIMENSION CNAME(12),ALE(12),SCM(12),CPC(36),GJET(200),Y(200),
     * UD(200),THD(200),TID(200),ED(200)
      REAL MOLF1(100),MOLF2(100),MOLF3(100),MOLF4(100),
     'MOLF5(100),MOLF6(100),MOLF7(100),MOLF8(100),
     'MOLF9(100),MOLF10(100),MOLF11(100),MOLF12(100)
      DIMENSION RAD(12),TS(12),UJ(12),SPV(12)
      INTEGER MWT(12)
      DIMENSION CP(12),FUEL(12),SPALDG(12),TKE(12),OTHER(36),
     'TITLE(20),PRINT(30),RCO(11),RCO2(11),RHC(11),RNOX(11),PT(11),
     'PS(11),BLOC(11),QCO(11),RICH(24),HCINCP(24),FUUL(48),ENTH(48),
     'CONC(768),OTHER2(192)
      DIMENSION SPLDG1(12),F(2,12,2),PDUM(20)
      EQUIVALENCE (FUUL(1),F(1,1,1))
C
C
      NAMELIST/A/YMIN,PDUM,YAIR
      NAMELIST/GMODOT/NPTS,YMIN,FUEL,F,SPALDG,SPLDG1,GJET,NJ,RICH
C
      DATA PDUM/20*0.0/
C
C
C     READ PREJET INPUT FILE
C
    1 READ (1)
     * DIAJ,MJET,TIJET,VJET,PE,TE,TIE,VE,GEX,RG,PR,PRT,SC,TREF,
     * MUREF,DIFF,NC,CNAME,ALE,SCM,CPC,NJ,NM,CT1,CT2,CT3,CT4,CT5,
     * CT6,CT7,CT8,GJET,Y,UD,THD,TID,ED,
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,
     * NPTS,RAD,TS,UJ,SPV,MWT,CP,FUEL,SPALDG,TKE,OTHER,
     * TITLE,PRINT,N,HF,WAR,T2,BETA,T25,FAR5,FINO2C,P0,
     * RCO,RCO2,RHC,RNOX,PT,PS,BLOC,QCO,
     * RICH,HCINCP,FUUL,ENTH,CONC,OTHER2
     * ,GEX
      REWIND 1
C
      YMIN=0.30
      YAIR=BITS
      READ(5,A)
C
      YMIN1=(1.0-YMIN)/YMIN
      DO 10 NN=1,NPTS
   10 SPLDG1(NN)=AMIN1(SPALDG(NN),YMIN1*(FUEL(NN)-F(2,NN,1))**2)
      GJET(1)=SPLDG1(1)
```

```
      NPP1=NPTS+1
      DO 50 NN=1,NPP1
      J=6*(NN-1)
      IF(NN.EQ.1)GO TO 20
      GJET(J)=0.5*(SPLDG1(NN-1)+SPLDG1(NN))
      GJET(J+1)=0.1*SPLDG1(NN-1)+0.9*SPLDG1(NN)
      IF(NN.EQ.NPP1)GO TO 40
   20 JP2=J+2
      JP4=J+4
      DO 30 J=JP2,JP4
   30 GJET(J)=SPLDG1(NN)
      GJET(JP4+1)=0.9*SPLDG1(NN)+0.1*SPLDG1(NN+1)
      GO TO 50
   40 GJET(J+2)=SPLDG1(NN)
   50 CONTINUE
C
      IF(YAIR.NE.BITS)RICH(NPP1)=YAIR
      IF(PDUM(1).NE.0.0)WRITE(6,GMODOT)
C
C
C     REWRITE PREJET FILE
C
      WRITE (1)
     * DIAJ,MJET,TIJET,VJET,PE,TF,TIF,VE,GFX,RG,PR,PRT,SC,TREF,
     * MUREF,DIFF,NC,CNAME,ALE,SCM,CPC,NJ,NM,CT1,CT2,CT3,CT4,CT5,
     * CT6,CT7,CT8,GJET,Y,UD,THD,TID,ED,
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,
     * NP1,RAD,T3,UJ,SPV,MW1,CP,FUEL,SPALDG,TKE,OTHER,
     * TITLE,PRINT,N,HF,WAR,T2,BETA,T25,FAR5,EINO2C,PO,
     * RCO,RCO2,RHC,RNOX,P1,PS,BLOC,QCO,
     * RICH,HCINCP,FUUL,ENTH,CONC,OTHER2
     * ,GEX
      RETURN
      END
```

```
CHCEQ2         FULL-EQUILIBRIUM COMPOSITION CH(N)-AIR-H2O SYSTEM
C              BRUTE-FORCE METHOD
C              LIKE HCEQ3 EXCEPT COMPOSITION RETURNED MOLE FRACTIONS
C
      SUBROUTINE HCEQ2(FAR,WAR,HC,T, P   ,FIXCO,FIXNO,ZZ)
C
      LOGICAL FIXCO,FIXNO,SAMIX,TROUBL,NOSOLN,DONE
      LOGICAL DISSOC
      REAL MF,K1,K2,K3,K4,K5,LNK6,LNPK6
      DIMENSION Z(11),ZZ(11)
      DIMENSION B(3),A(9),DZ(3)
      COMMON /GHSC  / FF(25),HH(25),SR(25),CPZ(25),DCPR(25)
C
      EQUIVALENCE(DSUMH,B(1)),(DSUMO,B(2)),(DPK6,B(3)),
     1(DZH2,DZ(1)),(DZO2,DZ(2)),(DZH2O,DZ(3))
C
C   ORDER OF SPECIES = H,O,H2,O2,OH,H2O,CO,CO2,N2,A,NO
C
      TROUBL= .FALSE.
      SAMIX=(FAR-OLDFAR+WAR-OLDWAR+HC-OLDHC).EQ.0.0
      IF(SAMIX)GO TO 10
C
C   CALCULATE EQUIVALENCE RATIO
C
      OLDFAR=FAR
      OLDWAR=WAR
      OLDHC=HC
      MF=12.01+1.008*HC
      FARS=0.209495/28.9666*MF/(1.0+0.25*HC)
      ER=FAR/FARS
C
C   TOTAL POUND-ATOMS EACH CONSTITUENT PER POUND MIX
C
      GAR=1.0+FAR+WAR
      SUMH=(FAR*HC/MF+WAR*2.0/18.016)/GAR
      SUMC=(FAR/MF+3E-4/28.9666)/GAR
      SUMO=(WAR/18.016+2.0*(0.209495+3E-4)/28.9666)/GAR
      SUMN=2.0*0.780881/28.9666/GAR
      SUMA=0.009324/28.9666/GAR
      CC=2.0*SUMC+0.5*SUMH-SUMO
      ZZX=0.5*(3E-4+28.9666/18.016*WAR)*SUMN/0.780881
      CCL=SUMC+0.5*SUMH-ZZX
      IF(CCL.LE.0.0)CCL=1E-15
      CCR=0.5*(SUMO-SUMC-ZZX)
   10 BETA1=0.01
      DPK6=0.0
C
C   SET CONCENTRATIONS OF ISOLATED SPECIES, ZERO OTHERS.
C
      CALL SETM(1,0.0,Z,11)
```

```
      Z(10)=SUMA
      IF(FIXCO)Z(7)=ZZ(7)/(ZZ(7)+ZZ(8))*SUMC
      IF(FIXNO)Z(11)=ZZ(11)/(ZZ(11)+2.0*ZZ(9))*SUMN
      IF(T.EQ.OLDT)GO TO 20
C
C   GET EQUILIBRIUM CONSTANTS
C
      OLDT=T
      TXK   = T /1.8
      CALL THRM( TXK , 1. )
      K1=EXP(FF(3)+FF(4)-2.0*FF(5))
      K2=EXP(FF(3)+FF(4)-FF(6)-FF(2))
      K3=FF(3)+FF(5)-FF(6)-FF(1)
      IF(K3.GT.88.0)K3=88.0
      K3=EXP(K3)
      K4=EXP(FF(8)+FF(3)-FF(6)-FF(7))
      K5=EXP(FF(9)+FF(4)-2.0*FF(11))
      LNK6=  FF(1)+FF(5)-FF(6)
   20 LNPK6=LNK6+ALOG(P/14.696)
      DISSOC=(LNPK6.LE.60.0).AND.(CCR.GT.1E-8)
      IF(.NOT.DISSOC)BETA1=0.0
C
C   INITIALIZE PRIMARY SPECIES
C
      IF(ER.GT.1.0)GO TO 30
      BETA1 = AMAX1( BETA1, Z(7)/CCL, (Z(11)+CC)/CCL )
      BETA1 = BETA1*1.0000001
      Z(3)=BETA1*CCL-Z(7)
      Z(4)=0.5*(Z(3)+Z(7)-Z(11)-CC)
      GO TO 40
   30 BETA1 = AMAX1( BETA1, 0.5*Z(11)/CCR, 0.5*(Z(7)-CC)/CCR )
      BETA1 = BETA1*1.0000001
      Z(4)=BETA1*CCR-0.5*Z(11)
      IF(FIXCO)GO TO 35
      AAA=K4-1.0
      CCC=CC+2.0*Z(4)+Z(11)
      BBB=-(CCC*AAA+K4*SUMC+0.5*SUMH)
      CCC=CCC*K4*SUMC
      Z(7)=-0.5*(BBB+SQRT(BBB**2-4.0*AAA*CCC))/AAA
   35 Z(3)=CC+2.0*Z(4)-Z(7)+Z(11)
   40 Z(6)=SUMO+Z(7)-Z(11)-2.0*(SUMC+Z(4))
      IF(DISSOC.AND.(Z(4).LE.0.0)) Z(4)=1E-30
C
C   BEGIN ITERATION
C
      DO 100 I=1,30
C
C   RADICALS FROM EQUILIBRIA
C
      Z(5)=SQRT(K1*Z(3)*Z(4))
```

```
      Z(2)=K2*Z(3)*Z(4)/Z(6)                                            HCEQ2101
      Z(1)=K3*Z(3)*Z(5)/Z(6)                                            HCEQ2102
      IF(FIXCO)GO TO 50                                                 HCEQ2103
      Z(7)=K4*SUMC*Z(3)/(Z(6)+K4*Z(3))                                  HCEQ2104
   50 Z(8)=SUMC-Z(7)                                                    HCEQ2105
      IF(FIXNO)GO TO 60                                                 HCEQ2106
      ZO2K5=0.0                                                         HCEQ2107
      IF(DISSOC)ZO2K5=0.25*K5*Z(4)                                      HCEQ2108
      Z(11)=SQRT(ZO2K5*(ZO2K5+2.0*SUMN))-ZO2K5                          HCEQ2109
   60 Z(9)=0.5*(SUMN-Z(11))                                             HCEQ2110
C                                                                       HCEQ2111
C   H,O CONTINUITY DEFECTS AND 3-BODY EQUILIBRIUM DEFECT                HCEQ2112
C                                                                       HCEQ2113
      DSUMH=SUMH-Z(1)-Z(5)-2.0*(Z(3)+Z(6))                              HCEQ2114
      DSUMO=SUMO-Z(2)-Z(5)-Z(6)-Z(7)-Z(11)-2.0*(Z(4)+Z(8))              HCEQ2115
      SPM=0.0                                                           HCEQ2116
      DO 70 J=1,11                                                      HCEQ2117
   70 SPM=SPM+Z(J)                                                      HCEQ2118
      IF(DISSOC)DPK6=LNPK6-ALOG(Z(6)/Z(1)*SPM/Z(5))                     HCEQ2119
C                                                                       HCEQ2120
C   TEST FOR CONVERGENCE                                                HCEQ2121
C                                                                       HCEQ2122
      DONE=(ABS(DSUMH).LT.1E-6*SUMH)                                    HCEQ2123
     '.AND.(ABS(DSUMO).LT.1E-6*SUMO)                                    HCEQ2124
     '.AND.(ABS(DPK6 ).LT.1E-6      )                                   HCEQ2125
      IF(DONE) GO TO 99                                                 HCEQ2126
C                                                                       HCEQ2127
C   PARTIAL DERIVATIVES TO DIRECT CONVERGENCE                           HCEQ2128
C                                                                       HCEQ2129
C                                                                       HCEQ2130
C   UNDISSOCIATED RICH MIXTURES                                         HCEQ2131
C                                                                       HCEQ2132
      IF(DISSOC)GO TO 75                                                HCEQ2133
      ZCODH2=Z(7)/Z(3)                                                  HCEQ2134
      ZCODC=Z(7)/SUMC                                                   HCEQ2135
      DZH2O=(DSUMO+0.5*ZCODH2*(1.0-ZCODC)*DSUMH)                        HCEQ2136
     '/(1.0+ZCODH2+ZCODH2*ZCODC*(1.0/K4-1.0))                           HCEQ2137
      DZH2=0.5*DSUMH-DZH2O                                              HCEQ2138
      DZO2=0.0                                                          HCEQ2139
      GO TO 99                                                          HCEQ2140
C                                                                       HCEQ2141
C                                                                       HCEQ2142
C   DISSOCIATED MIXTURES                                                HCEQ2143
C                                                                       HCEQ2144
   75 Z3=AMAX1(Z(3),1E-36)                                              HCEQ2145
      Z4=AMAX1(Z(4),1E-36)                                              HCEQ2146
      A(1)=2.0+0.5*(3.0*Z(1)+Z(5))/Z3                                   HCEQ2147
      A(2)=(Z(2)+0.5*Z(5))/Z3                                           HCEQ2148
      DCODX=K4*SUMC/(Z(6)+K4*Z(3))**2                                   HCEQ2149
      IF(.NOT.FIXCO)A(2)=A(2)-DCODX*Z(6)                                HCEQ2150
```

```
      A(3)=(1.5*Z(1)+Z(2)+Z(3)+0.5*Z(5)-2.0*SPM)/SPM/Z3                  HCEQ2151
      A(4)=0.5*(Z(1)+Z(5))/Z4                                            HCEQ2152
      A(5)=2.0+(Z(2)+0.5*Z(5))/Z4                                        HCEQ2153
      DNODO2=0.0                                                         HCEQ2154
      IF(Z(4).GT.4E-38/K5)DNODO2=0.5*K5*Z(9)/(0.25*K5*Z(4)+Z(11))        HCEQ2155
      IF(.NOT.FIXNO)A(5)=A(5)+DNODO2                                     HCEQ2156
      A(6)=0.5*(Z(1)+Z(5))+Z(2)+Z(4)-SPM                                 HCEQ2157
      IF(.NOT.FIXNO)A(6)=A(6)+0.5*DNODO2*Z(4)                            HCEQ2158
      A(6)=A(6)/SPM/Z4                                                   HCEQ2159
      A(7)=2.0-Z(1)/Z(6)                                                 HCEQ2160
      A(8)=1.0-Z(2)/Z(6)                                                 HCEQ2161
      IF(.NOT.FIXCO)A(8)=A(8)+DCODX*Z(3)                                 HCEQ2162
      A(9)=(-Z(1)-Z(2)+Z(6)+2.0*SPM)/SPM/Z(6)                            HCEQ2163
C                                                                        HCEQ2164
C   NEW ESTIMATES OF PRIMARY SPECIES                                     HCEQ2165
C                                                                        HCEQ2166
      CALL SOLV3(A,B,DZ,NOSOLN)                                          HCEQ2167
   99 TROUBL= TROUBL.OR.NOSOLN.OR. (I.EQ.20 )                            HCEQ2168
C                                                                        HCEQ2169
C   PROBLEM DIAGNOSTICS                                                  HCEQ2170
C                                                                        HCEQ2171
      IF(TROUBL)WRITE(6,1000)I,Z                                         HCEQ2172
 1000 FORMAT(1X,I2,8X,  1P11E11.4)                                       HCEQ2173
      IF(DONE)GO TO 110                                                  HCEQ2174
C                                                                        HCEQ2175
      DZLIM=-0.99*Z(3)                                                   HCEQ2176
      IF(DZH2.LT.DZLIM)DZH2=DZLIM                                        HCEQ2177
      Z(3)=Z(3)+DZH2                                                     HCEQ2178
      DZLIM=-0.99*Z(4)                                                   HCEQ2179
      IF(DZO2.LT.DZLIM)DZO2=DZLIM                                        HCEQ2180
      Z(4)=Z(4)+DZO2                                                     HCEQ2181
      Z(6)=Z(6)+DZH2O                                                    HCEQ2182
C                                                                        HCEQ2183
  100 CONTINUE                                                           HCEQ2184
      TROUBL= .TRUE.                                                     HCEQ2185
C                                                                        HCEQ2186
C   PROBLEM DIAGNOSTICS                                                  HCEQ2187
C                                                                        HCEQ2188
  110 IF(TROUBL)WRITE(6,1001)FAR,WAR,HC,T, P  ,FIXCO,FIXNO               HCEQ2189
 1001 FORMAT(15H0 I           H,                                         HCEQ2190
     '          10X,1HO,10X,2HH2,9X,2HO2,9X,2HOH,9X,3HH2O,8X,            HCEQ2191
     '2HCO,9X,3HCO2,8X,2HN2,9X,2HAR,9X,2HNO/                             HCEQ2192
     '25HOPSEQ2 DIAGNOSTICS   FAR=,F9.6,6H  WAR=,F9.6,5H  HC=,E10.3,     HCEQ2193
     '4H  T=,F9.2,7H     P=,F10.5 ,8H  FIXCO=,L2,8H  FIXNO=,L2)          HCEQ2194
C                                                                        HCEQ2195
C    COMPOSITION RETURNED AS MOLE FRACTIONS                              HCEQ2196
C                                                                        HCEQ2197
      CALL FMPYC(1,1.0/SPM,Z,ZZ,11)                                      HCEQ2198
C                                                                        HCEQ2199
      RETURN                                                             HCEQ2200
```

END HCEQ2201

```
CHCEQ3          FULL-EQUILIBRIUM COMPOSITION CH(N)-AIR-H2O SYSTEM
C            LIKE HCEQ2 EXCEPT COMPOSITION RETURNED MOLES/LB MIX
C
      SUBROUTINE HCEQ2(FAR,WAR,HC,T, P  ,FIXCO,FIXNO,ZZ)
C
      LOGICAL FIXCO,FIXNO,SAMIX,TROUBL,NOSOLN,DONE
      LOGICAL DISSOC
      REAL MF,K1,K2,K3,K4,K5,LNK6,LNPK6
      DIMENSION Z(11),ZZ(11)
      DIMENSION B(3),A(9),DZ(3)
      COMMON /GHSC  / FF(25),HH(25),SR(25),CPZ(25),DCPR(25)
C
      EQUIVALENCE(DSUMH,B(1)),(DSUMO,B(2)),(DPK6,B(3)),
     1(DZH2,DZ(1)),(DZO2,DZ(2)),(DZH2O,DZ(3))
C
C   ORDER OF SPECIES = H,O,H2,O2,OH,H2O,CO,CO2,N2,A,NO
C
      TROUBL= .FALSE.
      SAMIX=(FAR-OLDFAR+WAR-OLDWAR+HC-OLDHC).EQ.0.0
      IF(SAMIX)GO TO 10
C
C   CALCULATE EQUIVALENCE RATIO
C
      OLDFAR=FAR
      OLDWAR=WAR
      OLDHC=HC
      MF=12.01+1.008*HC
      FARS=0.209495/28.9666*MF/(1.0+0.25*HC)
      ER=FAR/FARS
C
C   TOTAL POUND-ATOMS EACH CONSTITUENT PER POUND MIX
C
      GAR=1.0+FAR+WAR
      SUMH=(FAR*HC/MF+WAR*2.0/18.016)/GAR
      SUMC=(FAR/MF+3E-4/28.9666)/GAR
      SUMO=(WAR/18.016+2.0*(0.209495+3E-4)/28.9666)/GAR
      SUMN=2.0*0.780881/28.9666/GAR
      SUMA=0.009324/28.9666/GAR
      CC=2.0*SUMC+0.5*SUMH-SUMO
      ZZX=0.5*(3E-4+28.9666/18.016*WAR)*SUMN/0.780881
      CCL=SUMC+0.5*SUMH-ZZX
      IF(CCL.LE.0.0)CCL=1E-15
      CCR=0.5*(SUMO-SUMC-ZZX)
   10 BETA1=0.01
      DPK6=0.0
C
C   SET CONCENTRATIONS OF ISOLATED SPECIES. ZERO OTHERS.
C
      CALL SETM(1,0.0,7,11)
      Z(10)=SUMA
```

```
      IF(FIXCO)Z(7)=ZZ(7)/(ZZ(7)+ZZ(8))*SUMC
      IF(FIXNO)Z(11)=ZZ(11)/(ZZ(11)+2.0*ZZ(9))*SUMN
      IF(T.EQ.OLDT)GO TO 20
C
C   GET EQUILIBRIUM CONSTANTS
C
      OLDT=T
      TXK   = T /1.8
      CALL THRM( TXK , 1. )
      K1=EXP(FF(3)+FF(4)-2.0*FF(5))
      K2=EXP(FF(3)+FF(4)-FF(6)-FF(2))
      K3=-FF(3)+FF(5)-FF(6)-FF(1)
      IF(K3.GT.88.0)K3=88.0
      K3=EXP(K3)
      K4=EXP(FF(8)+FF(3)-FF(6)-FF(7))
      K5=EXP(FF(9)+FF(4)-2.0*FF(11))
      LNK6=  FF(1)+FF(5)-FF(6)
   20 LNPK6=LNK6+ALOG(P/14.696)
      DISSOC= (LNPK6.LE.60.0).AND.(CCR.GT.1E-8)
      IF(.NOT.DISSOC)BETA1=0.0
C
C   INITIALIZE PRIMARY SPECIES
C
      IF(FR.GT.1.0)GO TO 30
      BETA1 = AMAX1( BETA1, Z(7)/CCL, (Z(11)+CC)/CCL )
      BETA1 = BETA1*1.0000001
      Z(3)=BETA1*CCL-Z(7)
      Z(4)=0.5*(Z(3)+Z(7)-Z(11)-CC)
      GO TO 40
   30 BETA1 = AMAX1( BETA1, 0.5*Z(11)/CCR, 0.5*(Z(7)-CC)/CCR )
      BETA1 = BETA1*1.0000001
      Z(4)=BETA1*CCR-0.5*Z(11)
      IF(FIXCO)GO TO 35
      AAA=K4-1.0
      CCC=CC+2.0*Z(4)+Z(11)
      BBB=-(CCC*AAA+K4*SUMC+0.5*SUMH)
      CCC=CCC*K4*SUMC
      Z(7)=-0.5*(BBB+SQRT(BBB**2-4.0*AAA*CCC))/AAA
   35 Z(3)=CC+2.0*Z(4)-Z(7)+Z(11)
   40 Z(6)=SUMO+Z(7)-Z(11)-2.0*(SUMC+Z(4))
      IF(DISSOC.AND.(Z(4).LE.0.0)) Z(4)=1E-30
C
C   BEGIN ITERATION
C
      DO 100 I=1,30
C
C   RADICALS FROM EQUILIBRIA
C
      Z(5)=SQRT(K1*Z(3)*Z(4))
      Z(2)=K2*Z(3)*Z(4)/Z(6)
```

```
      Z(1)=K3*Z(3)*Z(5)/Z(6)
      IF(FIXCO)GO TO 50
      Z(7)=K4*SUMC*Z(3)/(Z(6)+K4*Z(3))
   50 Z(8)=SUMC-Z(7)
      IF(FIXNO)GO TO 60
      ZO2K5=0.0
      IF(DISSOC)ZO2K5=0.25*K5*Z(4)
      Z(11)=SQRT(ZO2K5*(ZO2K5+2.0*SUMN))-ZO2K5
   60 Z(9)=0.5*(SUMN-Z(11))
C
C   H,O CONTINUITY DEFECTS AND 3-BODY EQUILIBRIUM DEFECT
C
      DSUMH=SUMH-Z(1)-Z(5)-2.0*(Z(3)+Z(6))
      DSUMO=SUMO-Z(2)-Z(5)-Z(6)-Z(7)-Z(11)-2.0*(Z(4)+Z(8))
      SPM=0.0
      DO 70 J=1,11
   70 SPM=SPM+Z(J)
      IF(DISSOC)DPK6=LNPK6-ALOG(Z(6)/Z(1)*SPM/Z(5))
C
C   TEST FOR CONVERGENCE
C
      DONE=(ABS(DSUMH).LT.1E-6*SUMH)
     1.AND.(ABS(DSUMO).LT.1E-6*SUMO)
     1.AND.(ABS(DPK6 ).LT.1E-6      )
      IF(DONE) GO TO 99
C
C   PARTIAL DERIVATIVES TO DIRECT CONVERGENCE
C
C   UNDISSOCIATED RICH MIXTURES
C
      IF(DISSOC)GO TO 75
      ZCODH2=Z(7)/Z(3)
      ZCODC=Z(7)/SUMC
      DZH2O=(DSUMO+0.5*ZCODH2*(1.0-ZCODC)*DSUMH)
     1/(1.0+ZCODH2+ZCODH2*ZCODC*(1.0/K4-1.0))
      DZH2=0.5*DSUMH-DZH2O
      DZO2=0.0
      GO TO 99
C
C   DISSOCIATED MIXTURES
C
   75 Z3=AMAX1(Z(3),1E-36)
      Z4=AMAX1(Z(4),1E-36)
      A(1)=2.0+0.5*(3.0*Z(1)+Z(5))/Z3
      A(2)=(Z(2)+0.5*Z(5))/Z3
      DCODX=K4*SUMC/(Z(6)+K4*Z(3))**2
      IF(.NOT.FIXCO)A(2)=A(2)-DCODX*Z(6)
      A(3)=(1.5*Z(1)+Z(2)+Z(3)+0.5*Z(5)-2.0*SPM)/SPM/Z3
      A(4)=0.5*(Z(1)+Z(5))/Z4
      A(5)=2.0+(Z(2)+0.5*Z(5))/Z4
```

```
      DNODO2=0.0                                                        HCFQ3151
      IF(Z(4).GT.4E-38/K5)DNODO2=0.5*K5*Z(9)/(0.25*K5*Z(4)+Z(11))       HCFQ3152
      IF(.NOT.FIXNO)A(5)=A(5)+DNODO2                                    HCFQ3153
      A(6)=0.5*(Z(1)+Z(5))+Z(2)+Z(4)-SPM                                HCFQ3154
      IF(.NOT.FIXNO)A(6)=A(6)+0.5*DNODO2*Z(4)                           HCFQ3155
      A(6)=A(6)/SPM/Z4                                                  HCFQ3156
      A(7)=2.0-Z(1)/Z(6)                                                HCFQ3157
      A(8)=1.0-Z(2)/Z(6)                                                HCFQ3158
      IF(.NOT.FIXCO)A(8)=A(8)+DCODX*Z(3)                                HCFQ3159
      A(9)=(-Z(1)-Z(2)+Z(6)+2.0*SPM)/SPM/Z(6)                           HCFQ3160
C                                                                       HCFQ3161
C    NEW ESTIMATES OF PRIMARY SPECIES                                   HCFQ3162
C                                                                       HCFQ3163
      CALL SOLV3(A,B,DZ,NOSOLN)                                         HCFQ3164
   99 TROUBL= TROUBL.OR.NOSOLN.OR. (I.EQ.20 )                           HCFQ3165
C                                                                       HCFQ3166
C    PROBLEM DIAGNOSTICS                                                HCFQ3167
C                                                                       HCFQ3168
      IF(TROUBL)WRITE(6,1000)I,Z                                        HCFQ3169
 1000 FORMAT(1X,I2,8X,  1P11E11.4)                                      HCFQ3170
      IF(DONE)GO TO 110                                                 HCFQ3171
C                                                                       HCFQ3172
      DZLIM=-0.99*Z(3)                                                  HCFQ3173
      IF(DZH2.LT.DZLIM)DZH2=DZLIM                                       HCFQ3174
      Z(3)=Z(3)+DZH2                                                    HCFQ3175
      DZLIM=-0.99*Z(4)                                                  HCFQ3176
      IF(DZO2.LT.DZLIM)DZO2=DZLIM                                       HCFQ3177
      Z(4)=Z(4)+DZO2                                                    HCFQ3178
      Z(6)=Z(6)+DZH2O                                                   HCFQ3179
C                                                                       HCFQ3180
  100 CONTINUE                                                          HCFQ3181
C                                                                       HCFQ3182
C    PROBLEM DIAGNOSTICS                                                HCFQ3183
C                                                                       HCFQ3184
  110 IF(TROUBL)WRITE(6,1001)FAR,WAR,HC,T, P   ,FIXCO,FIXNO             HCFQ3185
 1001 FORMAT(15H0 I              H,                                     HCFQ3186
     *                    10X,1HO,10X,2HH2,9X,2HO2,9X,2HOH,9X,3HH2O,8X,HCFQ3187
     *2HCO,9X,3HCO2,8X,2HN2,9X,2HAR,9X,2HNO/                            HCFQ3188
     *25HOPSEQ2 DIAGNOSTICS   FAR=,F9.6,6H  WAR=,F9.6,5H  HC=,E10.3,    HCFQ3189
     *4H  T=,F9.2,7H       P=,F10.5 ,8H  FIXCO=,L2,8H  FIXNO=,L2)       HCFQ3190
C                                                                       HCFQ3191
C    COMPOSITION RETURNED AS MOLES/LB MIX                               HCFQ3192
C                                                                       HCFQ3193
      CALL MOVE(1,Z,ZZ,11,1)                                            HCFQ3194
C                                                                       HCFQ3195
      RETURN                                                            HCFQ3196
      END                                                               HCFQ3197
```

```
CHYCARB         IGNITION DELAY FOR RAW FUEL                                 HYCARB01
      SUBROUTINE HYCARB(K)                                                  HYCARB02
      COMMON /INDATA/ N,HF,WAR,T2,BETA,T25,FAR5,EINO2C,P0,                  HYCARB03
     *                RCO(11),RCO2(11),RHC(11),RNOX(11),                    HYCARB04
     *                PT(11),PS(11),BLOC(11),QCO(11)                        HYCARB05
      REAL            N                                                     HYCARB06
      COMMON /JETDAT/ NPTS,DUM(108),TFR(12),TFL,TIM1(12),TIM2,OTHER1(10)    HYCARB07
      COMMON /GASCMP/ RICH(12,2),FUEL(2,12,2),ENTH(2,12,2),                 HYCARB08
     *                CONC(16,2,12,2),HCINCP(12,2),OTHER2(192)              HYCARB09
      COMMON /GASTMW/ TG(2,12,2),MWTG(2,12,2),TAU(12,2),CPG(2,12,2)         HYCARB10
      REAL            MWTG                                                  HYCARB11
      COMMON /OPCTRL/ TITLE(20),PRINT(30)                                   HYCARB12
      COMMON /STCTRL/ LSTA,FINAL,CHEMK,FIRSTM,FIRSTC,XC,DXC,DUMST(10)       HYCARB13
      LOGICAL         FINAL,FIRSTM,FIRSTC                                   HYCARB14
      INTEGER         CHEMK                                                 HYCARB15
      COMMON /CPRINT/ PDUM(20)                                              HYCARB16
      DATA A,B/-17.671439,40290.880/                                        HYCARB17
C                                                                           HYCARB18
C                                                                           HYCARB19
C     CALCULATE IGNITION DELAY                                              HYCARB20
C                                                                           HYCARB21
    1 DTIME = TIM2-TIM1(K)                                                  HYCARB22
      TEMP  = TG(1,K,1)                                                     HYCARB23
      DELAYI=1.E+30                                                         HYCARB24
      IF( TEMP.GE.400. ) DELAYI=.001*EXP(A+B/TEMP)                          HYCARB25
C                                                                           HYCARB26
C     WEIGHT INITIAL INCIPIENCIES                                           HYCARB27
C                                                                           HYCARB28
      HCI   = 0.                                                            HYCARB29
      TOTF  = 0.                                                            HYCARB30
      DO 5 KK=1,NPTS                                                        HYCARB31
      TOTF  = TOTF+TFR(KK)                                                  HYCARB32
      IF( HCINCP(KK,1).EQ. 1. ) GO TO 5                                     HYCARB33
      HCI   = HCI+TFR(KK)*HCINCP(KK,1)**4                                   HYCARB34
    5 CONTINUE                                                              HYCARB35
      TOTF  = TOTF+TFL                                                      HYCARB36
      IF( PDUM(13).NE.0.) CALL TABPRI(6HTFRETC,TFR,26,10)                   HYCARB37
      IF( TOTF.LE.0. ) GO TO 6                                              HYCARB38
      HCI   = (HCI/TOTF)**.25                                               HYCARB39
    6 HCINCP(K,2)= HCI+DTIME/DELAYI                                         HYCARB40
      IF( PDUM(13).NE.0.) WRITE (6,11) K,HCI,DELAYI,TEMP,HCINCP(K,2)        HYCARB41
   11 FORMAT(2X,I3,3X,4E16.8/)                                              HYCARB42
      IF( HCINCP(K,2).LE.1. ) GO TO 200                                     HYCARB43
C                                                                           HYCARB44
C     ADJUST CONC AND MW OF RICH FRACTION                                   HYCARB45
C                                                                           HYCARB46
C     REACTION-- C10H10N + 5O2 = 10CO + (5N)H2                              HYCARB47
C                                                                           HYCARB48
      C16   = CONC(16,1,K,2)                                                HYCARB49
      CONC(16,1,K,2)= 0.                                                    HYCARB50
```

```
      CONC(3,1,K,2)= CONC(3,1,K,2)+5.0*N*C16                            HYCARB51
      CONC(7,1,K,2)= CONC(7,1,K,2)+10.0*C16                             HYCARB52
      CONC(4,1,K,2)= CONC(4,1,K,2)-5.0*C16                              HYCARB53
      HCINCP(K,2)= 1.                                                   HYCARB54
  200 RETURN                                                            HYCARB55
      END                                                               HYCARB56
```

```
CINTEE      SET UP AND CALLFOR INTEGRATION
      SUBROUTINE INTE
C
C     SET UP AND CALL FOR INTEGRATION
C
C     OBTAIN SUGGESTED STEP SIZE FOR NEXT INTEGRATION STEP
C
      EXTERNAL PERR
C
      LOGICAL NEXT,ELIM
C
      REAL IVAR
C
      DIMENSION YN(33),Y(28),C(25)
C
      COMMON/OPTS/VERSI,TIMEV,VERSA,AREAV,FLIM,TCON,RHOCON,IPRCOD
      COMMON/COND/YNP1(33),LS,LSP3,NEXT
      COMMON/SINT/HMIN,HNM1,HN,HNP1,HMAX,NH,AVH,EMAX,ERRN,JCV,KOUNT,ERRP
      COMMON/PORE/PK(28),QK(28),RK(28),E(28)
      COMMON/SPEC/SNAM(2,30),MW(25),W(25),STOIC(25,30),OMEGA(25,30)
C
      EQUIVALENCE (IVAR,YNP1(5)),(Y,YNP1(6)),(C,YNP1(9)),(HINT,HNM1)
C
C     INITIAL STEPS OR RESTART
      ENTRY INTI
      NRFST = 0
      EMAX2 = EMAX/2.
      EMAX56 = 5.*EMAX/6.
      LSP8 = LS + 8
      HNP1 = HINT
      DO 3 I=1,LSP8
    3 YN(I) = YNP1(I)
    2 CALL CASI (HN,QK,HNP1,PK)
      HN = HNP1
      CALL PRED
      CALL CASI (HN,PK,HNP1,QK)
      HNM1 = HN
      HN = HNP1
      CALL PRED
      DO 23 I=1,LS
      IF (C(I) .GE. 0.) GO TO 23
      WRITE (6,100)
  100 FORMAT (7H0(INTE),5X,43HCOMPOSITION ERROR - NEGATIVE CONCENTRATION
     *S)
      NEXT = .TRUE.
      RETURN
   23 CONTINUE
      RETURN
C
C     PREPARE TO CONTINUE INTEGRATION
```

```
      ENTRY INTC                                                        INTEE051
      DO 4 I=1,LSP8                                                     INTEE052
    4 YNP1(I) = YN(I)                                                   INTEE053
      CALL PRED                                                         INTEE054
      RETURN                                                            INTEE055
C                                                                       INTEE056
C     GENERAL STEP                                                      INTEE057
      ENTRY INTG                                                        INTEE058
      CALL CASG (HN,QK,HNP1,RK)                                         INTEE059
C     RELATIVE ERROR IN INTEGRATION STEP                                INTEE060
      ERRN = ERRORC(Y,RK,F,JCV,HNP1)                                    INTEE061
C     TEST FOR RESTART CONDITIONS                                       INTEE062
      IF (ERRN .LE. EMAX  .OR.  HNP1 .GT. HMIN) GO TO 7                 INTEE063
C                                                                       INTEE064
      WRITE (6,101) IVAR,(SNAM(J,JCV),J=1,2),HNM1,HN,HNP1               INTEE065
  101 FORMAT (7H0(INTE),5X,7HRESTART,10X,22HINDEPENDENT VARIABLE =,E13.5INTEE066
     *,2X,11H(CGS UNITS),5X,23HCONTROLLING VARIABLE = ,2A4//51X,6HH(N-1)INTEE067
     *,17X,4HH(N),17X,6HH(N+1)/48X,3(E12.5,10X)//51X,6HK(N-1),17X,4HK(N)INTEE068
     *,17X,6HK(N+1),16X,6HE(N+1))                                       INTEE069
      DO 5 I=1,LSP8                                                     INTEE070
      IF (I .GT. LSP3) GO TO 5                                          INTEE071
      WRITE (6,102) (SNAM(J,I),J=1,2),PK(I),QK(I),RK(I),F(I)            INTEE072
  102 FORMAT (30X,2A4,4(10X,E12.5))                                     INTEE073
    5 YNP1(I) = YN(I)                                                   INTEE074
      NH = NH + 2                                                       INTEE075
C                                                                       INTEE076
C     STOP AFTER 10 RESTARTS                                            INTEE077
      IF (NREST .LT. 10) GO TO 6                                        INTEE078
      WRITE (6,103)                                                     INTEE079
  103 FORMAT (7H0(INTE),5X,25H10 RESTARTS HAVE OCCURRED)                INTEE080
      NEXT = .TRUE.                                                     INTEE081
      RETURN                                                            INTEE082
    6 NREST = NREST + 1                                                 INTEE083
      HMIN = HMIN/2.                                                    INTEE084
      HNP1 = HMIN                                                       INTEE085
      GO TO 2                                                           INTEE086
C                                                                       INTEE087
    7 HNM1 = HN                                                         INTEE088
      HN = HNP1                                                         INTEE089
      DO 8 I=1,LSP8                                                     INTEE090
      IF (I .GT. LSP3) GO TO 8                                          INTEE091
      PK(I) = QK(I)                                                     INTEE092
      QK(I) = RK(I)                                                     INTEE093
    8 YN(I) = YNP1(I)                                                   INTEE094
      CALL PRED                                                         INTEE095
C                                                                       INTEE096
C     TEST FOR NEGATIVE CONCENTRATIONS                                  INTEE097
      DO 9 I=1,LS                                                       INTEE098
      IF (C(I) .GE. 0.) GO TO 9                                         INTEE099
      WRITE (6,100)                                                     INTEE100
```

```
      NEXT = .TRUE.                                                     INTEE101
      RETURN                                                            INTEE102
    9 CONTINUE                                                          INTEE103
C                                                                       INTEE104
C     OPTIONAL AUTOMATIC ELIMINATIONS                                   INTEE105
      IF (ELIM) CALL AUTO                                               INTEE106
C                                                                       INTEE107
C     GET STEP SIZE FOR NEXT INTEGRATION STEP                           INTEE108
      KOUNT = 0                                                         INTEE109
      IF (ERRN .GE. EMAX2) GO TO 10                                     INTEE110
      IF (HN .GE. HMAX) RETURN                                          INTEE111
      CALL SEARCH (PERR,EMAX,HN,HNP1,HMAX)                              INTEE112
      RETURN                                                            INTEE113
   10 IF (ERRN .LE. EMAX56) RETURN                                      INTEE114
      IF (HN .LE. HMIN) RETURN                                          INTEE115
      CALL SEARCH (PERR,EMAX,HMIN,HNP1,HN)                              INTEE116
      RETURN                                                            INTEE117
C                                                                       INTEE118
      END                                                               INTEE119
```

```
CINTG          INTEGRAL OF Y*DX--TRAPEZOIDAL/UNEQUAL X S
      SUBROUTINE INTG(Y,X,YDXI,IL,IU)
      DIMENSION Y(1),X(1),YDXI(1)
C*
C*  EVALUATE INTEGRAL--- Y*DX  ***TRAPEZOIDAL RULE
C*        *******UNEQUAL SPACING
C*
    1 I=IL-1
    2 I=I+1
      DX=X(I)-X(I-1)
      TERM=.5*(Y(I)+Y(I-1))
    3 YDXI(I)=YDXI(I-1)+TERM*DX
      IF(I.LT. IU) GO TO 2
   10 RETURN
      END
```

```
CJETINP        JET--INPUT ROUTINE                                      JETINP01
      SUBROUTINE JETINP                                                JETINP02
C**********************************************************************JETINP03
C********************** SPECIAL VERSION FOR 12-SPECIES ****************JETINP04
C**********************************************************************JETINP05
      LOGICAL TAPIN,  TAPOT                                            JETINP06
      INTEGER PLOT                                                     JETINP07
      LOGICAL MCHANG                                                   JETINP08
      LOGICAL AMBIO                                                    JETINP09
      LOGICAL DPRIN                                                    JETINP10
      LOGICAL  EOF  ,  ERR                                             JETINP11
      LOGICAL AXI,CMPRS,QJET,TURBJ,CORE                                JETINP12
      INTEGER XPRN                                                     JETINP13
      LOGICAL ENTRY1                                                   JETINP14
      REAL  MJET , ME , MUREF                                          JETINP15
      REAL MACH                                                        JETINP16
      COMMON /FILK/CSC                                                 JETINP17
      COMMON /SING/  SSD(43)                                           JETINP18
      COMMON /CARRY/ NEW                                               JETINP19
      LOGICAL  NEW                                                     JETINP20
      COMMON /RSTART/ NREG,RESTRT,NRES,MIXPRE                          JETINP21
      LOGICAL  MIXPRE                                                  JETINP22
      COMMON /EDGE/  YJETE,  SFEDGE                                    JETINP23
      COMMON /JETTWO/ JTWO,ZDIAO(6),NJO                                JETINP24
      COMMON /BCO/  UO, EO, THO                                        JETINP25
      COMMON /CTRL2/                                                   JETINP26
     * EDGEI  , SFI    , MERGE  , XMERGE , YMERGE ,                    JETINP27
     * SLOPEI , SLOPEO , CEPTI  , CEPTO                                JETINP28
      COMMON /MERGET/  MER, MERSTP , XMRG                              JETINP29
      LOGICAL  TWO, MERGE , MER , MERSTP                               JETINP30
      COMMON /PROPJ2/ MACHO,REFLO,YI,YO,MERGP                          JETINP31
      REAL MACHO,MJETO                                                 JETINP32
      COMMON /MISC/ PM(10),  PLOT                                      JETINP33
      COMMON /INP1/ ENTRY1                                             JETINP34
      COMMON /UMESH/MCG(7)                                             JETINP35
C*                                                                     JETINP36
C****** INPUT COMMON                                                   JETINP37
C*                                                                     JETINP38
      COMMON /INPJET/                                                  JETINP39
     * ZDIAJ(11),JAXI,NJ,NM,X(100),XPRN(100),GAM,ZRG(6)                JETINP40
C*                                                                     JETINP41
C****** CONTROL COMMON                                                 JETINP42
C*                                                                     JETINP43
      COMMON /CTRL/                                                    JETINP44
     * NXTA        , CMPRS       , QJET        , TURBJ       , COEF(10)    , JETINP45
     * NPU  ,  NPO  , DXC  , XU  , XOO  ,                              JETINP46
     * DSTOR(800)                                                      JETINP47
C*                                                                     JETINP48
C****** PROFILE COMMON                                                 JETINP49
C*                                                                     JETINP50
```

```
      COMMON /PROF/ PSI(200),Y(200),UD(200),THD(200),ED(200)              JETINP51
C*                                                                        JETINP52
C****** CONSTANT AND ERROR COMMON                                         JETINP53
C*                                                                        JETINP54
      COMMON /CNERR/ BITS , ERR , GC , GCJ , FOOT                         JETINP55
C*                                                                        JETINP56
C****** BOUNDARY CONDITION COMMON                                         JETINP57
C*                                                                        JETINP58
      COMMON /BC/ UEDGE  , EEDGE  , THEDGE                                JETINP59
C*                                                                        JETINP60
C****** POTENTIAL CORE COMMON                                             JETINP61
C*                                                                        JETINP62
      COMMON /CORED/ XCORE  , CORE  , CORSTP                              JETINP63
C*                                                                        JETINP64
C****** SCALER (UNITS CONVERSION) COMMON                                  JETINP65
C*                                                                        JETINP66
      COMMON /SCALER/  SP  , SV  , SLEN                                   JETINP67
C*                                                                        JETINP68
C****** JET PROPERTIES COMMON                                             JETINP69
C*                                                                        JETINP70
      COMMON /JET/                                                        JETINP71
     * BB(100),UC(100),TC(100),TIC(100),                                  JETINP72
     * PTC(100) , WJ(100)  , YJ(100)  ,                                   JETINP73
     * YSONIC(100)                                                        JETINP74
      COMMON /JET1/ FLOWJ,TTO,NX,EJET                                     JETINP75
      COMMON /PROPJT/                                                     JETINP76
     * P       , PRL      , PRTT     , RGAS     , SCC      ,              JETINP77
     * TREF     , VSREF    , MACH     , XLC      ,                        JETINP78
     * XREFL,CDIFF,CHI,DUM(1401)                                          JETINP79
      COMMON /XPRIN/ DPRIN                                                JETINP80
      COMMON /CPROP/ CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,CT9,CT10             JETINP81
      COMMON /CPROP2/  CTP,  CTS  ,  CTM                                  JETINP82
      COMMON /RATIO/  AMBTO                                               JETINP83
      COMMON /IOFILE/  TAPIN,TAPOT                                        JETINP84
      COMMON /PARAM/ B(500),DIAJ(1),MJET,TJET,PTJET,VJET,                 JETINP85
     * TIJET,PE,VE,ME,TIE,TE,AXI,RG(1),PR,PRT,SC,TREF,                    JETINP86
     * MUREF,TWO,DIAO(1),MJETO,TJETO,VJETO,PTJETO,                        JETINP87
     * TIJETO,MCHANG(1),CK,DY1,NMSH,CXPC,CXTP,NRED,                       JETINP88
     * MIX,SUPR,MAXIT,TOL,CF,NB(1),TAB(5),ND(1),TAD(4),                   JETINP89
     * D(800),DUMINP(461)                                                 JETINP90
C*                                                                        JETINP91
      COMMON /MIXER/ JMIX,RD(100),XD(100),ZCF,YR(100)                     JETINP92
      LOGICAL  MIX                                                        JETINP93
      COMMON /FLOBAL/ MMAXIT,JSUP,NIT,PSID,YDD,YDC,                       JETINP94
     *               P1,P2,UCL,ZTOL,UPSTRM,CVG                            JETINP95
      LOGICAL  SUPR,CVG,UPSTRM                                            JETINP96
      COMMON /ACONVG/ YCD(100),PD(100),INDC(100),  CHOKE, CHOKED          JETINP97
      LOGICAL  CHOKE, CHOKED                                              JETINP98
      COMMON /DFIT/  CLSP(100)                                            JETINP99
      COMMON /STA2/ MACH2,TS2,SS2,V2,RHO2,DPDX2                           JETINP00
```

```
      REAL  MACH2                                                             JETINP01
      COMMON /BCMIX2/ GRADU,TW,MUW,RHOW,PTE,TIE                               JETINP02
      REAL  MUW                                                               JETINP03
      COMMON /CBODY/  YCB(100),CLSPCB(100),YCB1 , UCL1                        JETINP04
      COMMON /OUTMIX/ NXORIG                                                  JETINP05
      COMMON /INNAME/ MNB(6),MND(5)                                           JETINP06
      COMMON /TAG/  DUMID(40)  , IDENT(10)                                    JETINP07
      COMMON /DIFEQI/                                                         JETINP08
     * NC , CNAME(12) , ALJ(12) , ALJO(12) ,ALE(12) , SCM(12) ,               JETINP09
     * TCPRF(12) , HCPRF(12) , CPC(3,12)                                      JETINP10
      COMMON /DICTRL/ DIFF , CND(10)                                          JETINP11
      LOGICAL          DIFF                                                   JETINP12
      COMMON /MOLES / ALX(100,12)                                             JETINP13
      COMMON /BCMOL / ALEDGE(12),ALO(12)                                      JETINP14
      COMMON /JET3/  STADO,NV,STATF                                           JETINP15
      COMMON /LLTERP/ LTERP                                                   JETINP16
      LOGICAL         LTERP                                                   JETINP17
      COMMON /SCALED/ SCLD,ALXLIM                                             JETINP18
      LOGICAL         SCLD                                                    JETINP19
      COMMON /CRMOD / CJRMOD                                                  JETINP20
C*                                                                            JETINP21
      DIMENSION IMP(10)                                                       JETINP22
      EQUIVALENCE (IMP(1),PM(1))                                              JETINP23
      DIMENSION XPRN1(100)                                                    JETINP24
      EQUIVALENCE (XPRN1(1),XPRN(1))                                          JETINP25
      DIMENSION ICHANG(1)                                                     JETINP26
      DIMENSION DMOL(6)                                                       JETINP27
      EQUIVALENCE (ICHANG(1),MCHANG(1))                                       JETINP28
      EQUIVALENCE (IAXI,AXI),(ITWO,TWO),(ISUP,SUPB),(IMIX,MIX)                JETINP29
      EQUIVALENCE (IBITS,BITS)                                                JETINP30
      DIMENSION TID(200)                                                      JETINP31
      EQUIVALENCE (TID(1),ED(1))                                              JETINP32
      NAMELIST /A/ DIAJ,MJET,TJET,PTJET,VJET,                                 JETINP33
     * TIJET        ,                                                         JETINP34
     * PE           , VE           , ME           , TIE          , TE       , JETINP35
     * AXI          , NJ           , NM           ,                           JETINP36
     * GAM          , RG           , PR           , PRT          ,            JETINP37
     * X            , XPRN         ,                                          JETINP38
     * SC           , TREF         , MUREF        ,                           JETINP39
     * SP           , SV           , SLEN         ,                           JETINP40
     * DPRIN        ,PLOT          ,                                          JETINP41
     * NRED         , PM           ,                                          JETINP42
     * CXPC         , CXTP         ,                                          JETINP43
     * MCHANG       , IDENT       ,                                           JETINP44
     * Y  , UD , THD , ED , TID ,                                             JETINP45
     * CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,CT9,                                   JETINP46
     * CTM,CTS,CTP,B,NB,TAB,D,ND,TAD,RESTRT,                                  JETINP47
     * CK,DY1,NMSH,MIX,MAXIT,CF,TOL,SUPB,RD,XD,YCB,                           JETINP48
     * TWO       , DIAO       , MJETO      , TJETO      ,                     JETINP49
     * VJETO,PTJETO,TIJETO,NJO,                                               JETINP50
```

```
     * NC,CNAME,ALJ,ALJD,ALE,SCM,TCPRF,HCPRF,CPC,ALX,DIFF                  JETINP51
     * ,CSC,NDIGIT,LTERP,CDIFF,CHI,SCLD,ALXLIM , CJRMOD                    JETINP52
       DATA  BLANK/6H      /                                               JETINP53
C*                                                                         JETINP54
C*                                                                         JETINP55
C*   READ NAMELIST SA                                                      JETINP56
C*                                                                         JETINP57
    2 CALL FLGERR(5,ERR)                                                   JETINP58
      CALL SETM(1,BITS,Y,800)                                              JETINP59
      Y(1)=BITS                                                            JETINP60
      CALL SETM(1,BLANK,IDENT,10)                                          JETINP61
C*                                                                         JETINP62
C*   INITIALIZE B ARRAY, DUMMY SINGLE CELL INPUT                           JETINP63
C*                                                                         JETINP64
      IF(.NOT. NEW)  GO TO 4091                                            JETINP65
 4000 CALL SETM(2,BITS,B,543,ND,805)                                       JETINP66
      GO TO 4096                                                           JETINP67
 4091 CALL MOVE(1,SSD,DIAJ,43,1)                                           JETINP68
      NEW=.TRUE.                                                           JETINP69
      CALL SETM(5,BITS,ZDIAJ(2),4,ZDIAJ(8),2,ZDIAD(2),4,B,500,D,800)       JETINP70
 4096 IF(TAPIN)  CALL JTFILE(3,DUMYF)                                      JETINP71
    3 READ (5,A)                                                           JETINP72
      CALL MOVE(1,DIAJ,SSD,43,1)                                           JETINP73
  999 IF( ERR ) RETURN                                                     JETINP74
C*                                                                         JETINP75
C*   RESET POSSIBLE BITS VALUES                                            JETINP76
C*                                                                         JETINP77
      IF(IAXI.NE.IBITS) JAXI=IAXI                                          JETINP78
      IF(IMIX.NE.IBITS) JMIX=IMIX                                          JETINP79
      IF(ITWO.NE.IBITS) JTWO=ITWO                                          JETINP80
      IF(ISUP.NE.IBITS) JSUP=ISUP                                          JETINP81
      IF(MAXIT.NE.IBITS) MMAXIT=MAXIT                                      JETINP82
      IF(TOL.NE.BITS) ZTOL=TOL                                             JETINP83
      IF(CF.NE.BITS) ZCF=CF                                                JETINP84
      IF(CXPC.EQ.BITS .AND. TWO) CXPC=.04                                  JETINP85
      IF(CXTP.EQ.BITS .AND. TWO) CXTP=.04                                  JETINP86
      IF(CXPC.EQ.BITS .AND. MIX)  CXPC=.05                                 JETINP87
      IF(CXTP.EQ.BITS .AND. MIX)  CXTP=.05                                 JETINP88
 4001 DO 4010 L=1,11                                                       JETINP89
      IF(L.GT.6) GO TO 4002                                                JETINP90
      IF(RG(L).NE.BITS) ZRG(L)=RG(L)                                       JETINP91
      IF(DIAD(L).NE.BITS) ZDIAD(L)=DIAD(L)                                 JETINP92
 4002 IF(L.GT.7) GO TO 4003                                                JETINP93
      IF(ICHANG(L).NE.IBITS) MCG(L)=ICHANG(L)                              JETINP94
 4003 IF(DIAJ(L).NE.BITS) ZDIAJ(L)=DIAJ(L)                                 JETINP95
      IF(NB(L).NE.IBITS) MNB(L)=NB(L)                                      JETINP96
 4010 CONTINUE                                                             JETINP97
C*                                                                         JETINP98
C*   MOVE DATA BACK TO DUMMY ARRAY                                         JETINP99
C*                                                                         JETINP00
```

```
 4020 CALL MOVE(5,ZDIAJ,DIAJ,11,1,MCG,MCHANG,7,1,
     * ZDIAO,DIAO,6,1,ZRG,RG,6,1,MNB,NB,11,1)
      IAXI=JAXI
      ITWO=JTWO
      ISUP=JSUP
      IMIX=JMIX
      CF=ZCF
      TOL=ZTOL
      MAXIT=MMAXIT
      IF(MIX .AND. TWO)  GO TO 444
      IF(TWO)  CT1=.175
      IMP(1)=IFIX(PM(1))
      CMPRS=.FALSE.
      KGO=2
C*
C*  MOVE INPUT ARRAYS TO STORAGE
C*
 4030 CONTINUE
C*
C*  COUNT NUMBER OF AXIAL STATIONS
C*
C*
C*  IF MIXER NOZZLE CASE, NON-DIMENSIONALIZE
C*  AND CURVE FIT DUCT CO-ORDINATES
C*
      IF(RESTRT .NE. BITS) GO TO 4300
      IF(.NOT.MIX) GO TO 4442
      DO 4440 L=1,100
      IF(XD(L).EQ.BITS) GO TO 4445
 4440 CONTINUE
      ERR=.TRUE.
      GO TO 999
C*
 4445 NXTA=L-1
C*  FILL UNFILLED ARRAYS
C*
 4040 CALL FILL(XD,RD,1,NXTA)
      CALL FILL(XD,YCB,1,NXTA)
      TERMD=1./DIAJ
      CALL FMPYC(1,TERMD,XD,X,NXTA)
      TERMD=2./DIAJ
      CALL FMPYC(2,TERMD,RD,YR,NXTA,YCB,YCB,NXTA)
C*
      CALL LCFIT(XD,YR,NXTA,1,XD(1),Y99,1,0,CLSP)
      CALL LCFIT(XD,YCB,NXTA,1,XD(1),YCB1,1,0,CLSPCB)
      GO TO  6
 4442 DO 4 L=1,100
      IF(X(L).EQ.BITS) GO TO 5
    4 CONTINUE
  444 ERR=.TRUE.
```

```
      GO TO 999                                                          JETINP51
    5 NXTA=L-1                                                           JETINP52
      GO TO 6                                                            JETINP53
C*                                                                       JETINP54
C*  SPECIAL  X-TABLE PROCESSING ON RESTART                               JETINP55
C*                                                                       JETINP56
 4300 NXX=NXTA+1                                                         JETINP57
      NXD=NV+1                                                           JETINP58
      XLOC=RESTRT                                                        JETINP59
      IF(MIX)  RESTRT=RESTRT/DIAJ                                        JETINP60
C*                                                                       JETINP61
C*  SEARCH FOR RESTART LOC.                                              JETINP62
C*                                                                       JETINP63
      DO 4310 L=1,100                                                    JETINP64
      IF(X(L).EQ. RESTRT)  NRES=L                                        JETINP65
      IF(X(L).EQ.BITS) GO TO 4320                                        JETINP66
 4310 CONTINUE                                                           JETINP67
 4311 ERR=.TRUE.                                                         JETINP68
      GO TO 999                                                          JETINP69
 4320 NXTA=L-1                                                           JETINP70
      IF(NRES .EQ. 1) GO TO 4311                                         JETINP71
C*                                                                       JETINP72
      IF((.NOT. MIX) .AND. MIXPRE)  GO TO 4329                           JETINP73
      IF(.NOT. MIX)  GO TO 6                                             JETINP74
      DO 4340 L=1,100                                                    JETINP75
      IF(XD(L).EQ. BITS) GO TO 4350                                      JETINP76
 4340 CONTINUE                                                           JETINP77
 4350 NXORIG=L-1                                                         JETINP78
      CALL FILL(XD(NXD),RD(NXD),NXD,NXORIG)                              JETINP79
      CALL FILL(XD(NXD),YCB(NXD),NXD,NXORIG)                             JETINP80
      TERMD=1./DIAJ                                                      JETINP81
      NUM=NXORIG+1-NXD                                                   JETINP82
      CALL FMPYC(1,TERMD,XD(NXD),X(NXX),NUM)                             JETINP83
      TERMD=2.*TERMD                                                     JETINP84
      CALL FMPYC(2,TERMD,RD(NXD),YR(NXD),NUM,YCB(NXD),YCB(NXD),NUM)      JETINP85
      CALL LCFIT(XD,YR,NXORIG,1,XD(1),Y99,1,0,CLSP)                      JETINP86
      CALL LCFIT(XD,YCB,NXORIG,1,XD(1),YCB1,1,0,CLSPBC)                  JETINP87
      NXTA=NXX+NUM-1                                                     JETINP88
 4329 DO 4328 L=1,NXTA                                                   JETINP89
      IF(XLOC .EQ. XD(L))  PE=PD(L)                                      JETINP90
 4328 CONTINUE                                                           JETINP91
C*                                                                       JETINP92
C*  TEST FOR COMPRESSIBILITY,TRANSPORT OF Q--INITIALIZE                  JETINP93
C*                                                                       JETINP94
    6 IF(MJET.NE.BITS) CMPRS=.TRUE.                                      JETINP95
      IF(PTJET.NE. BITS) CMPRS=.TRUE.                                    JETINP96
      IF(.NOT. CMPRS) CALL SETM(1,BITS,PTC,100)                          JETINP97
      IF(MIX .AND. (RESTRT.EQ. BITS))  NXORIG=NXTA                       JETINP98
C*                                                                       JETINP99
C*  SET PRINT INDICATOR                                                  JETINP00
```

```
C*                                                                    JETINP01
 4050 IF((XPRN(1).EQ.2) .OR. (XPRN1(1).EQ.2.)) CALL SETM(1,1,XPRN(2), JETINP02
     *  NXTA-1)                                                       JETINP03
      IF((XPRN(1) .EQ. (-2))  .OR.  (XPRN1(1) .EQ. (-2.)))            JETINP04
     X  CALL SETM(1,-1,XPRN(2),NXTA-1)                                JETINP05
      DO 4060 L=2,NXTA                                                JETINP06
      IF(XPRN(L).EQ.IBITS) XPRN(L)=0                                  JETINP07
      IF(XPRN1(L) .EQ. 0.)  GO TO 4060                                JETINP08
      IF(IABS(XPRN(L)) .EQ. 1)  GO TO 4060                            JETINP09
      IF(XPRN1(L) .LT. 0.)  GO TO 4055                                JETINP10
      XPRN(L)=1                                                       JETINP11
      GO TO 4060                                                      JETINP12
 4055 XPRN(L)=-1                                                      JETINP13
 4060 CONTINUE                                                        JETINP14
      IF(TIJET.NE.0.) GO TO 10                                        JETINP15
      TURBJ=.FALSE.                                                   JETINP16
C*                                                                    JETINP17
C*                                                                    JETINP18
C*                                                                    JETINP19
C*                                                                    JETINP20
   10 IF(CMPRS)  GO TO 12                                             JETINP21
C*                                                                    JETINP22
C*  VJET,VE,TJET,TE   ASSUMED GIVEN                                   JETINP23
C*                                                                    JETINP24
      MJET=0.                                                         JETINP25
      ME=0.                                                           JETINP26
      CP=GAMH(TJET)                                                   JETINP27
      CPJ = CP                                                        JETINP28
      EJET=1.5/GCJ*TIJET**2*VJET**2                                   JETINP29
      IF(TWO)  EJETO=1.5/GCJ*TIJETO**2*VJET**2                        JETINP30
      TTE=TE                                                          JETINP31
      PTE=PE*(1.+.5*VE**2/(GC*RG*TE))                                 JETINP32
      GO TO 150                                                       JETINP33
   12 GAM=GAMH(TE)                                                    JETINP34
      VSE=SQRT(GAM*RG*GC*TE)                                          JETINP35
      IF(ME.EQ.BITS) GO TO 122                                        JETINP36
      VE=ME*VSE                                                       JETINP37
      GO TO 125                                                       JETINP38
  122 ME=VE/VSE                                                       JETINP39
  125 TRM=1.+.5*(GAM-1.)*ME**2                                        JETINP40
      TTE=TE*TRM                                                      JETINP41
      PTE=PE*(TRM)**(GAM/(GAM-1.))                                    JETINP42
      IF(MJET.EQ. BITS) KGO=1                                         JETINP43
      IF(PTJET.EQ.BITS) KGO=2                                         JETINP44
      GO TO (130,140),KGO                                             JETINP45
C*                                                                    JETINP46
  140 IF(TJET.NE.BITS) GO TO 141                                      JETINP47
C*                                                                    JETINP48
C*  DETERMINE TJET (GAM=GAM(T))                                       JETINP49
C*                                                                    JETINP50
```

```
      TLT=VJET*VJET/(RG*GC*MJET*MJET)
      TJET=(TLT/2.23708)**(1./.92979)
      IF(TJET.GT. 800.  .AND.  TJET.LT.3600.) GO TO 1410
      GAM=1.4
      IF(TJET.GE.3600.) GAM=1.254
      GO TO 1411
 1410 GAM=GAMH(TJET)
 1411 GAM1=GAM/(GAM-1.)
      GAM2=.5*(GAM-1.)
      GAM3=1./GAM1
      CP=GAM1*RG*GC/GCJ
      GO TO 142
  141 GAM=GAMH(TJET)
      VJET=MJET*SQRT(GAM*RG*GC*TJET)
      GO TO 1411
  142 EJET=1.5/GCJ*TIJET*TIJET*VJET*VJET
      PTJET=PE*(1.+GAM2*MJET**2)**GAM1
      GO TO 150
C*
  130 IF(VJET.EQ.BITS) GO TO 131
      ASSIGN 1312 TO KPGO
      GAMG  = 1.38
      IT    = 0
      PRAT  = PTJET/PE
 1301 IT    = IT+1
      GAMG1 = (GAMG-1.)/GAMG
      TJET  = VJET**2*(GAMG-1.)/(2.*GAMG*RG*GC*(PRAT**GAMG1-1.))
      IF( TJET.GE.800.  .AND.TJET.LT.3600. ) GO TO 1302
      GAM   = 1.4
      IF( TJET.GE.3600. ) GAM=1.254
      GO TO 1303
 1302 GAM   = GAMH(TJET)
 1303 IF( (ABS(GAM-GAMG)).LE. .001  .OR. IT.GE.10 ) GO TO 1311
      GAMG  = GAM
      GO TO 1301
 1310 GAM=GAMH(TJET)
 1311 GAM1=GAM/(GAM-1.)
      GAM2=.5*(GAM-1.)
      GAM3=1./GAM1
      CP=GAM1*RG*GC/GCJ
      TTRJ=(PTJET/PE)**GAM3
      GO TO KPGO,(1312,1364)
 1312 EJET=1.5/GCJ*TIJET*TIJET*VJET*VJET
      MJET=VJET/SQRT(GAM*RG*GC*TJET)
      GO TO 150
  131 ASSIGN 1364 TO KPGO
      GO TO 1310
 1364 MJET=SQRT((TTRJ-1.)/GAM2)
      VJET=MJET*SQRT(GAM*RG*GC*TJET)
      EJET=1.5/GCJ*TIJET*TIJET*VJET*VJET
```

```
C*
  150 CPJ=CP
C*
      IF(.NOT. TWO) GO TO 151
C**  LOGIC FOR FLOW CONDITIONS OF OUTER JET
C**
      IF(CMPRS)  GO TO 160
      MJETO=0.
      GO TO 151
  160 IF(MJETO.EQ.BITS)  KHO=1
      IF(PTJETO.EQ.BITS)  KHO=2
      GO TO  (170,165)  ,  KHO
  165 IF(TJETO.NE.BITS)  GO TO 168
C*
C*  DETERMINE TJETO (GAMO=GAMO(T))
C*
      TLT=VJETO*VJETO/(RG*GC*MJETO*MJETO)
      TJETO=(TLT/2.23708)**(1./.92979)
      IF(TJETO.GT.800.  .AND.  TJETO.LT.3600.) GO TO 1610
      GAMO=1.4
      IF(TJETO.GE.3600.) GAM=1.254
      GO TO 1611
 1610 GAMO=GAMH(TJETO)
 1611 GAM1=GAM/(GAM-1.)
      GAM2=.5*(GAM-1.)
      GAM3=1./GAM1
      CP=GAM1*RG*GC/GCJ
      GO TO 169
  168 GAMO=GAMH(TJETO)
      VJETO=MJETO*SQRT(GAMO*RG*GC*TJETO)
      GO TO 1611
  169 EJETO=1.5/GCJ*TIJETO*TIJETO*VJET*VJET
      PTJETO=PE*(1.+GAM2*MJETO**2)**GAM1
      GO TO  151
C*
C*
  170 IF(VJETO.EQ. BITS) GO TO 171
      ASSIGN 1712 TO KPGO
      GAMG  = 1.38
      IT    = 0
      PRAT  = PTJETO/PE
 1701 IT    = IT+1
      GAMG1 = (GAMG-1.)/GAMG
      TJETO = VJETO**2*(GAMG-1.)/(2.*GAMG*RG*GC*(PRAT**GAMG1-1.))
      IF( TJETO.GE.800. .AND. TJETO.LT.3600. ) GO TO 1702
      GAMO  = 1.4
      IF( TJETO.GE.3600. ) GAMO=1.254
      GO TO 1703
 1702 GAMO  = GAMH(TJETO)
 1703 IF( (ABS(GAMG-GAMO)).LE. .001  .OR. IT.GE.10 ) GO TO 1711
```

```
      GAMG  = GAMO                                                      JETINP51
      GO TO 1701                                                        JETINP52
 1710 GAMO=GAMH(TJETO)                                                  JETINP53
 1711 GAM1=GAMO/(GAMO-1.)                                               JETINP54
      GAM2=.5*(GAMO-1.)                                                 JETINP55
      GAM3=1./GAM1                                                      JETINP56
      CP=GAM1*RG*GC/GCJ                                                 JETINP57
      TTRJO=(PTJETO/PE)**GAM3                                           JETINP58
      GO TO KPGO,(1712,1764)                                            JETINP59
 1712 EJETO=1.5/GCJ*TIJETO*TIJETO*VJET*VJET                             JETINP60
      MJETO=VJETO/SQRT(GAMO*RG*GC*TJETO)                                JETINP61
      GO TO 151                                                         JETINP62
  171 ASSIGN 1764 TO KPGO                                               JETINP63
      GO TO 1710                                                        JETINP64
 1764 EJETO=1.5/GCJ*TIJETO*TIJETO*VJET*VJET                             JETINP65
      MJETO=SQRT((TTRJO-1.)/GAM2)                                       JETINP66
      VJETO=MJETO*SQRT(GAM*RG*GC*TJETO)                                 JETINP67
  151 IF( TWO)  GO TO 152                                               JETINP68
      IF(TJET.EQ.TE)  QJET=.FALSE.                                      JETINP69
      GO TO 153                                                         JETINP70
  152 IF(TJET.EQ.TE   .AND.   TJET.EQ.TJETO)  QJET=.FALSE.              JETINP71
  153 CONTINUE                                                          JETINP72
      IF(.NJT. QJET) CALL SETM(1,1.,TC,100)                             JETINP73
      EE=1.5/GCJ*TIE*TIE*VJET*VJET                                      JETINP74
      TTO=TJET+.5*VJET*VJET/(GCJ*CPJ)+EJET/CPJ                          JETINP75
      AMBTO=.FALSE.                                                     JETINP76
      DTBO=ABS(TTO-TE)                                                  JETINP77
      IF(DTBO.LT.   .5) AMBTO=.TRUE.                                    JETINP78
      CALL MOVE(3,DIAJ,ZDIAJ,6,1,DIAJ,ZDIAJ,11,1,MCHANG,MCG,7,1)        JETINP79
C*                                                                      JETINP80
C*  INITIALIZE PROFILES                                                 JETINP81
C*                                                                      JETINP82
   20 NJP=NJ+1                                                          JETINP83
      NJM=NJ-1                                                          JETINP84
C*                                                                      JETINP85
C*  BOUNDARY CONDITIONS--V/VJET=VE/VJET,THETA=0,                        JETINP86
C*  E/EJET=EE/EJET                                                      JETINP87
C*                                                                      JETINP88
   23 THEDGE=TE/TJET                                                    JETINP89
      IF(TURBJ) EEDGE=EE/EJET                                           JETINP90
      UEDGE=VE/VJET                                                     JETINP91
C*                                                                      JETINP92
C*  MESH DEFINITION AT INITIAL STATION-- IF Y(1).NE.BITS,               JETINP93
C                                                                       JETINP94
C*  INPUT PROFILES USED AS GIVEN                                        JETINP95
C*  NJ=CURRENT SPECIFIED MESH NUMBER OF JET CORNER                      JETINP96
C*                                                                      JETINP97
      U0=1.                                                             JETINP98
      TH0=1.                                                            JETINP99
      IF(TURBJ)  E0=1.                                                  JETINP00
```

```
      DO 2310 LL=1,NC                                                   JETINP01
      ALEDGE(LL)=ALE(LL)                                                JETINP02
 2310 ALO(LL)=ALJ(LL)                                                   JETINP03
      IF(.NOT. TWO)  GO TO 2313                                         JETINP04
      THO=TJETO/TJET                                                    JETINP05
      UO=VJETO/VJET                                                     JETINP06
      IF(TURBJ)  EO=EJETO/EJET                                          JETINP07
      DO 2311 LL=1,NC                                                   JETINP08
 2311 ALO(LL)=ALJO(LL)                                                  JETINP09
 2313 IGO=1                                                             JETINP10
      IF(RESTRT.NE.BITS) GO TO 3840                                     JETINP11
      IF(Y(1).NE.BITS) IGO=2                                            JETINP12
      GO TO (21,30), IGO                                                JETINP13
C*                                                                      JETINP14
C*  GENERATE INITIAL PROFILES                                           JETINP15
C*                                                                      JETINP16
   21 LGO=1                                                             JETINP17
      LOW=2                                                             JETINP18
      NE=NJ+3                                                           JETINP19
      Y(1)=0.                                                           JETINP20
      IF(MIX) Y(1)=YCB1                                                 JETINP21
      UD(1)=1.                                                          JETINP22
      THD(1)=1.                                                         JETINP23
      IF(TURBJ)  ED(1)=1.                                               JETINP24
C*                                                                      JETINP25
      DYI=(1.-Y(1))/FLOAT(NJ-1)                                         JETINP26
      DY2=.5*DYI                                                        JETINP27
      LHO=1                                                             JETINP28
      IF(TWO)  LHO=2                                                    JETINP29
      IF(.NOT.DIFF) GO TO 2161                                          JETINP30
      DO 2160 LL=1,NC                                                   JETINP31
 2160 ALX(1,LL) = ALJ(LL)                                               JETINP32
 2161 GO TO (22,2222),LHO                                               JETINP33
   22 DU=UEDGE-UO                                                       JETINP34
      DTH=THEDGE-THO                                                    JETINP35
      IF(TURBJ)  DE=EEDGE-EO                                            JETINP36
      IF(.NOT.DIFF) GO TO 26                                            JETINP37
      DO 2170 LL=1,NC                                                   JETINP38
 2170 DMOL(LL)=ALEDGE(LL)-ALO(LL)                                       JETINP39
      GO TO 26                                                          JETINP40
 2222 DU=UO-1.                                                          JETINP41
      DTH=THO-1.                                                        JETINP42
      IF(TURBJ)  DE=EO-1.                                               JETINP43
      IF(.NOT.DIFF) GO TO 26                                            JETINP44
      DO 2171 LL=1,NC                                                   JETINP45
 2171 DMOL(LL)=ALO(LL)-ALJ(LL)                                          JETINP46
   26 DO 27 L=LOW,NE                                                    JETINP47
      IF(L.GT.NJM) GO TO 25                                             JETINP48
      Y(L)=Y(L-1)+DYI                                                   JETINP49
      UD(L)=UD(LOW-1)                                                   JETINP50
```

```
      THD(L)=THD(LOW-1)                                                 JETINP51
      IF(.NOT.TURBJ) GO TO 2619                                         JETINP52
      ED(L)=ED(LOW-1)                                                   JETINP53
 2619 IF(.NOT.DIFF) GO TO 27                                            JETINP54
      DO 2620 LL=1,NC                                                   JETINP55
 2620 ALX(L,LL)=ALX(LOW-1,LL)                                           JETINP56
      GO TO 27                                                          JETINP57
   25 Y(L)=Y(L-1)+DY2                                                   JETINP58
      UD(L)=UD(L-1)+.25*DU                                              JETINP59
      THD(L)=THD(L-1)+.25*DTH                                           JETINP60
      IF(.NOT.TURBJ) GO TO 2519                                         JETINP61
      ED(L)=ED(L-1)+.25*DE                                              JETINP62
 2519 IF(.NOT.DIFF) GO TO 27                                            JETINP63
      DO 2520 LL=1,NC                                                   JETINP64
 2520 ALX(L,LL)=ALX(L-1,LL)+.25*DMOL(LL)                                JETINP65
   27 CONTINUE                                                          JETINP66
      GO TO  (32,2223)  ,  LHO                                          JETINP67
 2223 DIST=DIAO/DIAJ-(1.+DYI)                                           JETINP68
      DYI=DIST/FLOAT(NJO-NE)                                            JETINP69
      EDGEI=Y(NE)                                                       JETINP70
      LOW=NE+1                                                          JETINP71
      NE=NJO+3                                                          JETINP72
      NJM=NJO-1                                                         JETINP73
      NJP=NJO+1                                                         JETINP74
      LHO=1                                                             JETINP75
      GO TO 22                                                          JETINP76
   32 NRMN=NM-NE                                                        JETINP77
   31 NE1=NE+1                                                          JETINP78
      CALL SETM(1,UEDGE,UD(NE1),NRMN)                                   JETINP79
      CALL SETM(1,THEDGE,THD(NE1),NRMN)                                 JETINP80
      IF(TURBJ) CALL SETM(1,EEDGE,ED(NE1),NRMN)                         JETINP81
      BB(1)=Y(NE)-Y(NJM)                                                JETINP82
      IF(.NOT. DIFF) GO TO 3119                                         JETINP83
      DO 3120 LL=1,NC                                                   JETINP84
      ALE1=ALEDGE(LL)                                                   JETINP85
      CALL SETM(1,ALE1,ALX(NE1,LL),NRMN)                                JETINP86
 3120 CONTINUE                                                          JETINP87
 3119 GO TO (29,40),LGO                                                 JETINP88
   29 DO 28 L=NE1,NM                                                    JETINP89
   28 Y(L)=Y(L-1)+DYI                                                   JETINP90
      NPU=NE                                                            JETINP91
      NPD=NE                                                            JETINP92
      UD(NPU)=UEDGE                                                     JETINP93
      THD(NPU)=THEDGE                                                   JETINP94
      IF(TURBJ) ED(NPU)=EEDGE                                           JETINP95
      IF(.NOT.DIFF) GO TO 35                                            JETINP96
      DO 2802 LL=1,NC                                                   JETINP97
 2802 ALX(NPU,LL)=ALEDGE(LL)                                            JETINP98
      GO TO 35                                                          JETINP99
C*                                                                      JETINP00
```

```
C*    PROFILES INPUT BY USER                                            JETINP01
C*                                                                      JETINP02
   30 DO 36 L=1,NM                                                      JETINP03
      IF( Y(L).EQ.BITS ) GO TO 37                                       JETINP04
   36 CONTINUE                                                          JETINP05
      ERR    = .TRUE.                                                   JETINP06
      GO TO 999                                                         JETINP07
   37 L      = L-1                                                      JETINP08
      NE     = L                                                        JETINP09
      LGO    = 2                                                        JETINP10
      CALL FILL(Y,UD,1,NE)                                              JETINP11
      CALL FILL(Y,TID,1,NE)                                             JETINP12
      CALL FILL(Y,THD,1,NE)                                             JETINP13
C*    PROFILES  DIMENSIONAL-- IN.,FPS.,DEG. R.                          JETINP14
      QD     = 2./DIAJ                                                  JETINP15
      QV     = 1./VJET                                                  JETINP16
      QT     = 1./TJET                                                  JETINP17
      CALL FMPYC(1,QD,Y,Y,NE)                                           JETINP18
      CALL FMPYC(1,QV,UD,UD,NE)                                         JETINP19
      CALL FMPYC(1,QT,THD,THD,NE)                                       JETINP20
C                                                                       JETINP21
C     CALCULATE TKE FROM INPUT TURBULENT INTENSITY                      JETINP22
C                                                                       JETINP23
      TKET   = 1.5*VJET*VJET/(GCJ*EJET)                                 JETINP24
      DO 38 L=1,NE                                                      JETINP25
   38 ED(L) = TKET*TID(L)**2                                            JETINP26
      GO TO 32                                                          JETINP27
C*                                                                      JETINP28
C*                                                                      JETINP29
C*   WIDTH OF MIXING ZONE-- Y(NJP)-Y(NJM))---- 3-MESH POINTS            JETINP30
C*                                                                      JETINP31
   40 YJETE = Y(NJ)                                                     JETINP32
      UCCL1 = UD(1)                                                     JETINP33
      IF( SCLD ) GO TO 4111                                             JETINP34
      DO 41 L=2,NJ                                                      JETINP35
      IF( UD(L).NE.UCCL1 ) GO TO 42                                     JETINP36
   41 CONTINUE                                                          JETINP37
      ERR    = .TRUE.                                                   JETINP38
      GO TO 999                                                         JETINP39
 4111 DO 4112 L=1,NJ                                                    JETINP40
      IF( ALX(L,NC).GE.ALXLIM ) GO TO 42                                JETINP41
 4112 CONTINUE                                                          JETINP42
      ERR    = .TRUE.                                                   JETINP43
      GO TO 999                                                         JETINP44
   42 BB(1) = YJETE-Y(L-1)                                              JETINP45
      NPU=NE                                                            JETINP46
      NPD=NE                                                            JETINP47
      GO TO 35                                                          JETINP48
 3840 CALL JTFILE (4,RESTRT)                                            JETINP49
      NPU=NPD                                                           JETINP50
```

```
C*
C*
C*  COMPUTE CONSTANT TERMS IN COFFICIENTS OF PDEQS
C*
   35 DIA=DIAJ/12.
      YJETF=Y(NPU)
      UCL1=UD(1)
      IF(RESTRT.NE. BITS)  YJETE=YJ(NRES)
C*
      COEF(1)=1./(VJET*DIA)
      IF(TURBJ) GO TO 770
      COEF(2)=0.
      GO TO 769
  770 COEF(2)=VJET/(GCJ*DIA*EJET)
  769 IF(QJET) GO TO 771
      COEF(3)=0.
      COEF(4)=0.
      GO TO 772
  771 COEF(3)=EJET/(VJET*DIA*TJET)
      COEF(4)=VJET/(GCJ*DIA*TJET)
  772 COEF(5)=DIA/VJET
      COEF(6)=.5*DIA
      COEF(7)=144.*GC/(VJET*VJET)
      COEF(9)=144.*GC/(GCJ*TJET)
C*
C*  INITIALIZE COMMON /PROPJ/
C*
      RGAS=RG
      IF(.NOT.DIFF) GO TO 7772
      WMWT=1545./RGAS
      CND(1)=1./(WMWT*VJET*DIA*TJET)
 7772 MACH=MJET
      P=PE
      VSREF=MUREF
      TRFF=TREF
      SCC=SC
      PRL=PR
      PRTT=PRT
      MACHO=MJETO
      ENTRY1=.TRUE.
      IF(RESTRT .NE.BITS) GO TO 996
      CALL JETPRP
C*
C*
C*  PRINT INITIAL STATION DATA
C*
  996 CALL JTOUT1
      IF(TAPOT .AND. RESTRT.EQ. BITS) CALL JTFILE(1,X(NRES))
C*
 1000 RETURN
```

```
      END                                                                 JETINP01
```

```
*JETPRF      GENFRATES INITIAL PROFILES FOR -JETMIX-
      SUBROUTINE JETPRF
      COMMON /OPCTRL/ TITLE(20),PRINT(30)
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
      LOGICAL          ERR,ERRMAJ,INERR,PRERR
      COMMON /JETDAT/ NPTS,RAD(12),TS(12),U(12),SPV(12),MWT(12),CP(12),
     1  FUEL(12),SPALDG(12),TKE(12),OTHER(36)
      COMMON /INDATA/ N,HF,WAR,T2,BETA,T25,FAR5,EINO2C,P0,RCO(11),
     1  RCO2(11),RHC(11),RNOX(11),PT(11),PS(11),BLOC(11),QCO(11)
      COMMON /CREFDA/
     * DIAJ,MJET,TIJET,VJET,GJET(200),PE,TF,TIE,VE,GFX,RG,PR,
     1  PRT,SC,TREF,MUREF,DIFF,NC,CNAME(12),ALF(12),SCM(12),CPC(36),NJ,
     * NM,CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,Y(200),UD(200),THD(200),
     * TID(200),ALX(1200)
      COMMON /KEYS/ KEYA(11),KEYB(11),KODA(11),KODB(11)
      COMMON /CPCOEF/ CA,CB,CC,CAA,CBA,CCA
      COMMON /GASCMP/ RICH(24),FUUL(48),ENTH(48),CONC(768),HCINCP(24),
     *                OTHER2(192)
      COMMON /CBITS / BITS,BLANK
      DIMENSION LEN1(27),NSA(26)
      DIMENSION RTUBE(12),CNAM1(12)
      DIMENSION ED(200)
      EQUIVALENCE (ED(1),TID(1))
      REAL      MOLF1(100),MOLF2(100),MOLF3(100),MOLF4(100),
     *          MOLF5(100),MOLF6(100),MOLF7(100),MOLF8(100),
     *          MOLF9(100),MOLF10(100),MOLF11(100),MOLF12(100)
      EQUIVALENCE  (MOLF1,ALX(1)),(MOLF2,ALX(101)),(MOLF3,ALX(201)),
     *             (MOLF4,ALX(301)),(MOLF5,ALX(401)),(MOLF6,ALX(501)),
     *             (MOLF7,ALX(601)),(MOLF8,ALX(701)),(MOLF9,ALX(801)),
     *           (MOLF10,ALX(901)),(MOLF11,ALX(1001)),(MOLF12,ALX(1101))
      REAL N,MASS,MACH,MJET,MUREF
      LOGICAL DIFF
      DIMENSION UJ(12)
      EQUIVALENCE (UJ,U)
      NAMELIST /A/ TIJET,TIE,RG,PR,PRT,SC,TREF,MUREF,NM,
     * CA,CB,CC,CAA,CBA,CCA,CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8
C*
      DATA LEN1/3,12,1,36,6,100,12,100,9,12,1,36,1,20,1,30,
     * 8,11,2,24,2,48,1,768,1,192,0/
      DATA NSA/3,1HI,1,1HJ,6,1HK,1,1HL,9,1HM,1,1HN,
     *         1,2HI1,1,2HJ1,8,2HK1,2,2HL1,2,2HM1,1,2HN1,1,2HI2/
      DATA CNAM1 /4HGAS1,4HGAS2,4HGAS3,4HGAS4,4HGAS5,4HGAS6,4HGAS7,
     1  4HGAS8,4HGAS9,5HGAS10,5HGAS11,5HGAS12/
C*
C*
      CALL FLGERR( 5,INERR)
      READ (5,A)
      IF( INERR ) RETURN
      CALL SETM(1,BITS,Y,200)
      CALL MOVE (1,CNAM1,CNAME,12,1)
```

```
      RJET=12.*RAD(NPTS+1)                                              JETPRF51
      DIAJ=2.*RJET                                                      JETPRF52
      PE=PO                                                             JETPRF53
      TE=TS(NPTS+1)                                                     JETPRF54
      VE=U(NPTS+1)                                                      JETPRF55
      GEX=SPALDG(NPTS+1)                                                JETPRF56
      NC=NPTS+1                                                         JETPRF57
      NJ=6*NPTS-1                                                       JETPRF58
      DIFF=.TRUE.                                                       JETPRF59
      MJET=0.                                                           JETPRF60
      VJET=0.                                                           JETPRF61
      FLOW=0.                                                           JETPRF62
      RTUBE(1)=0.                                                       JETPRF63
      PI=3.1415927                                                      JETPRF64
      GC=32.174                                                         JETPRF65
      DO 120 I=1,NPTS                                                   JETPRF66
      IF (I.EQ.NPTS) GO TO 100                                          JETPRF67
      RTUBE(I+1)=SQRT((RAD(I)**2+RAD(I+1)**2)/2.)                       JETPRF68
      GO TO 110                                                         JETPRF69
  100 RTUBE(I+1)=RAD(NPTS+1)                                            JETPRF70
  110 AREA=PI*(RTUBE(I+1)**2-RTUBE(I)**2)                               JETPRF71
      MASS=AREA*U(I)/SPV(I)                                             JETPRF72
      FLOW=FLOW+MASS                                                    JETPRF73
      GAMMA=1.4                                                         JETPRF74
      IF (TS(I).GT.(800.)) GAMMA=2.23708/TS(I)**.070271                 JETPRF75
      IF (TS(I).GE.(3600.)) GAMMA=1.254                                 JETPRF76
      MACH=U(I)/SQRT(GAMMA*GC*RG*TS(I))                                 JETPRF77
      MJET=MJET+MASS*MACH                                               JETPRF78
      VJET=VJET+MASS*U(I)                                               JETPRF79
  120 CONTINUE                                                          JETPRF80
      MJET=MJET/FLOW                                                    JETPRF81
      VJET=VJET/FLOW                                                    JETPRF82
      DO 130 I=1,NPTS                                                   JETPRF83
      CPC(3*I-2)=CA                                                     JETPRF84
      CPC(3*I-1)=CB                                                     JETPRF85
      CPC(3*I)=CC                                                       JETPRF86
      SCM(I)=.70                                                        JETPRF87
      ALE(I)=0.0                                                        JETPRF88
  130 CONTINUE                                                          JETPRF89
      NPP1=NPTS+1                                                       JETPRF90
      CPC(3*NPP1-2)=CAA                                                 JETPRF91
      CPC(3*NPP1-1)=CBA                                                 JETPRF92
      CPC(3*NPP1)=CCA                                                   JETPRF93
      SCM(NPP1)=.70                                                     JETPRF94
      ALE(NPP1)=1.0                                                     JETPRF95
      NSL=NJ+3                                                          JETPRF96
      NJM1=NJ-1                                                         JETPRF97
      CALL SETM (1,TIJET,TID(1),NJM1)                                   JETPRF98
      NALX=100*NPP1                                                     JETPRF99
      CALL SETM (1,0.0,ALX(1),NALX)                                     JETPRF00
```

```
      DO 150 J=NJ,NSL                                                   JETPRF01
      IF (J.EQ.NJ) SCL=.90                                              JETPRF02
      IF (J.EQ.NJ+1) SCL=.50                                            JETPRF03
      IF (J.EQ.NJ+2) SCL=.10                                            JETPRF04
      IF (J.EQ.NSL) SCL=0.0                                             JETPRF05
      TID(J)=SCL*TIJET+(1.-SCL)*TIE                                     JETPRF06
  150 CONTINUE                                                          JETPRF07
      DRSAVE=.010*RAD(NPTS+1)                                           JETPRF08
      Y(1)=12.*RTUBE(1)                                                 JETPRF09
      UD(1) = U(1)                                                      JETPRF10
      THD(1)= TS(1)                                                     JETPRF11
      GJET(1)= SPALDG(1)                                                JETPRF12
      ALX(1)=1.                                                         JETPRF13
      DO 200 I=1,NPP1                                                   JETPRF14
      J      = 6*(I-1)                                                  JETPRF15
      DR=DRSAVE                                                         JETPRF16
      DRQ2=DR/2.                                                        JETPRF17
      IF (I.EQ.NPP1) GO TO 160                                          JETPRF18
      IF (DR.LT.(RTUBE(I+1)-RTUBE(I))/2.) GO TO 160                     JETPRF19
      DR=(RTUBE(I+1)-RTUBE(I))/4.                                       JETPRF20
      DRQ2=DR/2.                                                        JETPRF21
  160 IF (I.EQ.1) GO TO 170                                             JETPRF22
      Y(J)=12.*RTUBE(I)                                                 JETPRF23
      UD(J) = 0.5*(U(I-1)+U(I))                                         JETPRF24
      THD(J)= .5*(TS(I-1)+TS(I))                                        JETPRF25
      GJET(J)= .5*(SPALDG(I-1)+SPALDG(I))                               JETPRF26
      K=100*(I-2)+J                                                     JETPRF27
      ALX(K)=.50                                                        JETPRF28
      K=100*(I-1)+J                                                     JETPRF29
      ALX(K)=.50                                                        JETPRF30
      JP1=J+1                                                           JETPRF31
      Y(JP1)=12.*(RTUBE(I)+DRQ2)                                        JETPRF32
      UD(JP1) = 0.10*U(I-1)+0.90*U(I)                                   JETPRF33
      THD(JP1)=.10*TS(I-1)+.90*TS(I)                                    JETPRF34
      GJET(JP1)= .1*SPALDG(I-1)+.9*SPALDG(I)                            JETPRF35
      K=100*(I-2)+JP1                                                   JETPRF36
      ALX(K)=.10                                                        JETPRF37
      K=100*(I-1)+JP1                                                   JETPRF38
      ALX(K)=.90                                                        JETPRF39
      IF (I.EQ.NPP1) GO TO 190                                          JETPRF40
  170 Y(J+2)= 12.*(RTUBE(I)+DR)                                         JETPRF41
      Y(J+3)=12.*(RTUBE(I)+RTUBE(I+1))/2.                               JETPRF42
      Y(J+4)=12.*(RTUBE(I+1)-DR)                                        JETPRF43
      JP2=J+2                                                           JETPRF44
      JP4=J+4                                                           JETPRF45
      DO 180 J=JP2,JP4                                                  JETPRF46
      UD(J) = U(I)                                                      JETPRF47
      THD(J)=TS(I)                                                      JETPRF48
      GJET(J)= SPALDG(I)                                                JETPRF49
      K=100*(I-1)+J                                                     JETPRF50
```

```
      ALX(K)=1.0                                                        JETPRF51
  180 CONTINUE                                                          JETPRF52
      J=JP4+1                                                           JETPRF53
      Y(J)=12.*(RTUBE(I+1)-DRO2)                                        JETPRF54
      UD(J) = 0.90*U(I)+0.10*U(I+1)                                     JETPRF55
      THD(J)=.90*TS(I)+.10*TS(I+1)                                      JETPRF56
      GJET(J)= .9*SPALDG(I)+.1*SPALDG(I+1)                              JETPRF57
      K=100*(I-1)+J                                                     JETPRF58
      ALX(K)=.90                                                        JETPRF59
      K=100*I+J                                                         JETPRF60
      ALX(K)=.10                                                        JETPRF61
      GO TO 200                                                         JETPRF62
  190 J=J+2                                                             JETPRF63
      Y(J)=12.*(RTUBE(I)+DR)                                            JETPRF64
      UD(J) = U(I)                                                      JETPRF65
      THD(J)=TS(I)                                                      JETPRF66
      GJET(J)= SPALDG(I)                                                JETPRF67
      K=100*(I-1)+J                                                     JETPRF68
      ALX(K)=1.0                                                        JETPRF69
  200 CONTINUE                                                          JETPRF70
      LEN1(2)=NPP1                                                      JETPRF71
      LEN1(4)=3*NPP1                                                    JETPRF72
      LEN1(6)=NSL                                                       JETPRF73
      LEN1(8)= NSL                                                      JETPRF74
      LEN1(10)=NPP1                                                     JETPRF75
      LEN1(12)=1                                                        JETPRF76
      WRITE (1)                                                         JETPRF77
     * DIAJ,MJET,TIJET,VJET,PE,TE,TIE,VE,GEX,RG,PR,PRT,SC,TREF,         JETPRF78
     * MUREF,DIFF,NC,CNAME,ALF,SCM,CPC,NJ,NM,CT1,CT2,CT3,CT4,CT5,       JETPRF79
     * CT6,CT7,CT8,GJET,Y,UD,THD,TID,ED,                                JETPRF80
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,                             JETPRF81
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,                          JETPRF82
     * NPTS,RAD,TS,UJ,SPV,MWT,CP,FUEL,SPALDG,TKE,OTHER,                 JETPRF83
     * TITLE,PRINT,N,HF,WAR,T2,BF1A,T25,FAR5,EINO2C,P0,                 JETPRF84
     * RCO,RCO2,RHC,RNOX,PT,PS,BLOC,QCO,                                JETPRF85
     * RICH,HCINCP,FUUL,ENTH,CONC,OTHER2                                JETPRF86
     * ,GEX                                                             JETPRF87
      RETURN                                                            JETPRF88
      END                                                               JETPRF89
```

```
CJETPRP          PROPERTIES OF JET                                      JETPRP01
      SUBROUTINE JETPRP                                                 JETPRP02
      LOGICAL SUPERT                                                    JETPRP03
      LOGICAL ENTRY1                                                    JETPRP04
      LOGICAL  AXI , XPRN , CMPRS , QJET , TURBJ , CORE                 JETPRP05
      LOGICAL SUBSON,TMACH                                              JETPRP06
      LOGICAL ERR                                                       JETPRP07
      LOGICAL AMBTO                                                     JETPRP08
      REAL  MJET , ME , MUREF                                           JETPRP09
      REAL MACH                                                         JETPRP10
      REAL KCP,MUEFF,MUL                                                JETPRP11
      COMMON /JETTWO/                                                   JETPRP12
     * TWO     , DIAO    , MJETO   , TJETO   , VJETO   ,                JETPRP13
     * PTJETO  , TIJETO  , NJO                                          JETPRP14
      REAL  MJETO,MACHO                                                 JETPRP15
      COMMON /BCO/  UO, EO, THO                                         JETPRP16
      COMMON /CTRL2/                                                    JETPRP17
     * EDGEI  , SEI    , MERGE  , XMERGE , YMERGE ,                     JETPRP18
     * SLOPEI , SLOPEO , CEPTI  , CEPTO                                 JETPRP19
      COMMON /MERGET/  MER, MERSTP , XMRG                               JETPRP20
      LOGICAL  TWO, MERGE , MER , MERSTP                                JETPRP21
      COMMON /PROPJ2/ MACHO,REFLO,YI,YO,MERGP                           JETPRP22
      LOGICAL MERGP                                                     JETPRP23
      COMMON /RATIO/   AMBTO                                            JETPRP24
      COMMON /INP1/ ENTRY1                                              JETPRP25
C****** INPUT COMMON                                                    JETPRP26
C*                                                                      JETPRP27
      COMMON /INPJET/                                                   JETPRP28
     * DIAJ        , MJET        , TJET        , PTJET       , VJET        , JETPRP29
     * TIJET       ,                                                    JETPRP30
     * PE          , VE          , ME          , TIE         , TE          , JETPRP31
     * AXI         , NJ          , NM          ,                        JETPRP32
     * X(100)      , XPRN(100)   ,                                      JETPRP33
     * GAM         , RG          , PR          , PRT         ,          JETPRP34
     * SC          , TREF        , MUREF                                JETPRP35
C*                                                                      JETPRP36
C****** CONTROL COMMON                                                  JETPRP37
C*                                                                      JETPRP38
      COMMON /CTRL/                                                     JETPRP39
     * NXTA        , CMPRS       , QJET        , TURBJ       , COEF(10)    , JETPRP40
     * NPU  ,  NPD  , DXC  , XU  , XDD  ,                               JETPRP41
     * DSTOR(800)                                                       JETPRP42
C*                                                                      JETPRP43
C****** PROFILE COMMON                                                  JETPRP44
C*                                                                      JETPRP45
      COMMON /PROF/ PSI(200),YD(200),UD(200),THD(200),ED(200)           JETPRP46
C*                                                                      JETPRP47
C****** CONSTANT AND ERROR COMMON                                       JETPRP48
C*                                                                      JETPRP49
      COMMON /CNERR/ BITS , ERR , GC , GCJ , FOOT                       JETPRP50
```

```
C*                                                                      JETPRP51
C****** BOUNDARY CONDITION COMMON                                       JETPRP52
C*                                                                      JETPRP53
      COMMON /BC/ UEDGE  , EEDGE  , THEDGE                              JETPRP54
C*                                                                      JETPRP55
C****** POTENTIAL CORE COMMON                                            JETPRP56
C*                                                                      JETPRP57
      COMMON /CORES/ XCORE  , CORE  , CORSTP                            JETPRP58
C*                                                                      JETPRP59
C****** SCALER (UNITS CONVERSION) COMMON                                 JETPRP60
C*                                                                      JETPRP61
      COMMON /SCALER/  SP  , SV  , SLEN                                 JETPRP62
C*                                                                      JETPRP63
C****** JET PROPERTIES COMMON                                           JETPRP64
C*                                                                      JETPRP65
      COMMON /JET/                                                      JETPRP66
     * B(100)   , UC(100)   , TC(100)   , TIC(100) ,                    JETPRP67
     * PTC(100) , WJ(100)   , YJ(100)   ,                               JETPRP68
     * YSONIC(100)                                                      JETPRP69
      COMMON /JET1/ FLOWJ,TTO,NX,EJET                                   JETPRP70
      COMMON /JET2/  TTC(100)                                           JETPRP71
      COMMON /PROPJT/                                                   JETPRP72
     * P        , PRL       , PRTT      , RGAS      , SCC      ,        JETPRP73
     * TREFF    , VSREF     , MACH      , XLC       ,                   JETPRP74
     * REFL     , C         , CHI       , RNORM     ,                   JETPRP75
     * RHO(200)    , MUL(200)    , KCP(200)     ,                       JETPRP76
     * MUEFF(200) , XLN(200)    , DK(200)     , RETURB(200)             JETPRP77
      COMMON /EDGE/   YJETE   ,   SFEDGE                                JETPRP78
C*                                                                      JETPRP79
      COMMON /MIXER/ MIX,RD(100),XD(100),CF,YR(100)                     JETPRP80
      LOGICAL  MIX                                                      JETPRP81
      COMMON /FLOBAL/ MAXIT,SUPB,NIT,PSID,YDD,YDC,                      JETPRP82
     *                     P1,P2,UCL,TOL,UPSTRM,CVG                     JETPRP83
      LOGICAL  SUPB,CVG,UPSTRM                                          JETPRP84
      COMMON /ACONVG/ YCD(100),PD(100),INDC(100),  CHOKE, CHOKED        JETPRP85
      LOGICAL  CHOKE, CHOKED                                            JETPRP86
      COMMON /DFTT/  CLSP(100)                                          JETPRP87
      COMMON /STA2/ MACH2,TS2,SS2,V2,RHO2,DPDX2                         JETPRP88
      REAL  MACH2                                                       JETPRP89
      COMMON /BCMIX2/ GRADU,TW,MUW,RHOW,PIF,TIF                         JETPRP90
      REAL  MUW                                                         JETPRP91
      COMMON /THRST/WV(100)                                             JETPRP92
      COMMON /MIXPRP/ MA2(100),VE2(100),TE2(100),TWC(100)               JETPRP93
      REAL MA2                                                          JETPRP94
      COMMON /THERM/ GMC(200),CP(200)                                   JETPRP95
      COMMON /OUTMIX/ NXORIG                                            JETPRP96
      COMMON /CBODY/  YCB(100),CLSPCB(100),YCB1 , UCL1                  JETPRP97
      COMMON /JET3/ STADD, NV, STATF                                    JETPRP98
      COMMON /TNNAME/ NB, TAB(5) , ND , TAD(4)                          JETPRP99
      COMMON /CONSTF/ CON1                                              JETPRP00
```

```
      LOGICAL  STADD,STATF
      COMMON /SCALED/ SCLD,ALXLIM
      LOGICAL         SCLD
      COMMON /MOLES/ ALX(100,12)
      COMMON /DIFEQI/
     * NC , CNAME(12) , ALJ(12) , ALJO(12) , ALE(12) ,
     * SCM(12) , TCPRF(12) , HCPRF(12) , CPC(3,12)
      COMMON /YOFXI / IE
      DIMENSION T(200), STOR1(200)
      DIMENSION Y(200)
      DIMENSION TKE(200),STOR2(200)
      EQUIVALENCE (TKE(1),DSTOR(601)),(STOR2(1),DSTOR(601))
      DIMENSION YMACH(200)
      EQUIVALENCE (YMACH(1),DSTOR(601))
      EQUIVALENCE (Y(1),DSTOR(401))
      EQUIVALENCE (T(1),DSTOR(1)), (STOR1(1), DSTOR(201))
      EQUIVALENCE (C6,COEF(6))
      EQUIVALENCE (ITWO,TWO)
      DATA NREG/1/
C*   PROPERTIES DIMENSIONLESS UNLESS OTHERWISE NOTED
C*
    1 IF(.NOT.ENTRY1) GO TO 10
      ITWOJ=ITWO+1
      NX=1
      NV=1
      DTO=TJET-TF
      XJ=X(1)
      IGO=1
C*   INITIALIZE AT 1-ST STATION
C*
    2 UC(1)=UD(1)
      TC(1)=THD(1)
      TIC(1)=TIJET
      PTC(1)=1.
      TTC(1)=1.
      IF(AMBTO) TTC(1)=BITS
      IF(MJET.EQ. 0.) PTC(1)=BITS
      YJ(1)=YD(NPD)
C*   EVALUATE MASS FLOW IN JET--STREAM FUNCTION
C*
    3 EXP=1.
      IF(AXI) EXP=2.
      CON1=VJET*C6**EXP
      REFL=B(1)*C6
   99 CALL FMPYC(1,C6,YD,Y,NPD)
      CALL FMPYC(1,EJET,ED,TKE,NPD)
      CALL FMPYC(1,TJET,THD,T,NPD)
      IF( SCLD ) GO TO 98
      CALL SCALE(UD,ITWOJ,1,X(1))
      GO TO 97
```

```
   98 CALL SCALE(ALX(1,NC),ITWOJ,1,X)
   97 CALL PROPJ(ITWOJ,TURBJ,1,XJ,Y,T,TKE,1,NPD)
      CALL GAMCP(T,GMC,CP,RG,1,NPD)
      DO 5 L=1,NPD
      RAD=1.
      IF(AXI) RAD=YD(L)
    5 STOR1(L)=.5*RHO(L)*UD(L)*RAD
    6 PSI(1)=0.
   66 CALL INTG(STOR1,YD,PSI,2,NPD)
C*
C*  IF MIXER NOZZLE CASE, COMPUTE STREAM
C*  FUNCTION COINCIDENT WITH DUCT WALL
C*
      IF(.NOT. MIX) GO TO 666
      YO=YR(1)
      YI=YD(NPD)
      RHOE=TJET*RHO(1)/TE
      DPSIE=.5*RHOE*VE/(EXP*VJET)*(YO**EXP-YI**EXP)
      PSID=PSI(NPD)+DPSIE
      PD(1)=PE
      MA2(1)=ME
      VE2(1)=VE
      TE2(1)=TE
      YCD(1)=BITS
      TWC(1)=BITS
  666 WJ(1)=1.
      FLOWJ=PSI(NPD)*CON1
      CPJ=GAM*RG/((GAM-1.)*GC/GCJ
      SUBSON=.TRUE.
      IF(ME.GE. 1. .OR. MJET .GE. 1.) SUBSON=.FALSE.
      GO TO 8760
C*
C*  ENTRIES AT STATIONS OTHER THAN STATION 1
C*
   10 UC(NX)=UD(1)
      TC(NX)=THD(1)
      TIC(NX)=SQRT(2.*EJET*GCJ*ED(1)/3.)/VJET
C*
   11 YJ(NX)=YJETE
      YSET=YJETE
      IF(TWO) YSET=EDGEI
      IF( NREG.NE.2 ) NREG=1
      IF( .NOT.SCLD ) GO TO 1110
      IF( ALX(1,NC).GE.ALXLIM ) NREG=2
      GO TO 1111
 1110 IF( UD(2).NE.UD(3)) NREG=2
 1111 GO TO (12,14) , NREG
C*
C*  POTENTIAL CORE REGION
C*
```

```
   12 DO 13 L=1,NPD                                                     JETPRP01
      IF( SCLD ) GO TO 1200                                             JETPRP02
      IF(UD(L).NE.UCL1)  GO TO 15                                       JETPRP03
      GO TO 13                                                          JETPRP04
 1200 IF( ALX(L,NC).GE.ALXLIM ) GO TO 15                                JETPRP05
   13 CONTINUE                                                          JETPRP06
   15 B(NX)=YSET-YD(L-1)                                                JETPRP07
      GO TO 20                                                          JETPRP08
   14 B(NX)=2.*YJETE                                                    JETPRP09
C*                                                                      JETPRP10
C*  CL TOTAL PRESSURE                                                   JETPRP11
C*                                                                      JETPRP12
C*                                                                      JETPRP13
   20 IF(.NOT. CMPRS) GO TO 22                                          JETPRP14
      TEMP=THD(1)*TJET                                                  JETPRP15
      VJC=VJET*UD(1)                                                    JETPRP16
      GAM=GMC(1)                                                        JETPRP17
      CPJ=CP(1)                                                         JETPRP18
      GAM=GAMH(TL)                                                      JETPRP19
      TTCL=TEMP+.5*VJC*VJC/(GCJ*CPJ)+EJET*ED(1)/CPJ                     JETPRP20
      IF(.NOT.  AMBTO) GO TO 1337                                       JETPRP21
      TTC(NX)=BITS                                                      JETPRP22
      GO TO 1338                                                        JETPRP23
 1337 TTC(NX)=(TTCL-TE)/(TTO-TE)                                        JETPRP24
 1338 CONTINUE                                                          JETPRP25
      XMJ=VJC/SQRT(GAM*GC*RG*TEMP)                                      JETPRP26
      TTRJ=TTCL/TEMP                                                    JETPRP27
      PK=PE                                                             JETPRP28
      IF(MIX) PK=P2                                                     JETPRP29
      POC=PK*(1.+.5*(GAM-1.)*XMJ**2)**(GAM/(GAM-1.))                    JETPRP30
      PTC(NX)=(POC-PE)/(PTJET-PE)                                       JETPRP31
C*                                                                      JETPRP32
C*  LOCATE SONIC LINE IF SUPERSONIC***                                  JETPRP33
C*                                                                      JETPRP34
 8760 IF(SUBSON)    GO TO 29                                            JETPRP35
      TMACH=.FALSE.                                                     JETPRP36
      SUPERT=.TRUE.                                                     JETPRP37
      LK=2                                                              JETPRP38
      IE    = 1                                                         JETPRP39
      DO 211 L=1,NPD                                                    JETPRP40
      IF(L.GT. 1  .AND.  UD(L).EQ. UCL1)  IK=L                          JETPRP41
      TL=THD(L)*TJET                                                    JETPRP42
      GAM=GAMH(TL)                                                      JETPRP43
      VSON=SQRT(GAM*GC*RG*TL)                                           JETPRP44
      YMACH(L)=UD(L)*VJET/VSON                                          JETPRP45
      IF(L.LT.NPD .AND. YMACH(L).EQ. YMACH(L-1))  LK=L                  JETPRP46
      IF(.NOT. SUPERT)  GO TO 217                                       JETPRP47
      SUPERT= YMACH(L).LT. 1.                                           JETPRP48
  217 IF(TMACH)  GO TO 211                                              JETPRP49
      TMACH=YMACH(L).GE.1.                                              JETPRP50
```

```
  211 CONTINUE
      IF(SUPERT)  GO TO 29
      IF(.NOT. YMACH) SUBSON=.TRUE.
  212 IF(.NOT. SUBSON) GO TO 213
      YSONIC(NX)=0.
      GO TO 30
  213 YSONIC(NX)=YOF(1.,YMACH,YD,LK,NPD-1)
      IF(.NOT. ERR) GO TO 30
      ERR=.FALSE.
   29 YSONIC(NX)=BITS
      GO TO 30
   22 PTC(NX)=BITS
      TTC(NX)=BITS
C*
C*
C*  EVALUATE MASS FLOW IN JET
C*
   30 IF(NX.EQ.1) GO TO 1000
      WJ(NX)=SFEDGE/FLOWJ*CON1
C*
C*
C*   STORE PARAMETERS FOR CONFINED JET
C*
 1000 IF(.NOT. MIX) GO TO 1001
      IF(STADD) GO TO 1001
      IF(NX .EQ. 1) GO TO 996
      YCD(NV)=YDC
      PD(NV)=P2
      IF(.NOT. SUBSON)  INDC(NV)=1
      IF(CHOKE) INDC(NV)=2
      MA2(NV)=MACH2
      VE2(NV)=V2
      TE2(NV)=TS2
C*
C*  IF MIXER NOZZLE CASE,  CALCULATE THRUST
C*
  996 CALL  THRUST(NV)
C*
C*  IF FLOW CHOKED, CHECK AVAILABLE AREA AT NEXT STATION.
C*  IF   .LE. CURRENT AREA TERMINATE
C*
      IF(.NOT. CHOKE)  GO TO 1001
      A1=YDD**EXP-YCB1**EXP
      IF(NV.EQ. NXORIG)  GO TO 1001
      NV1=NV+1
      XNX=XD(NV1)
      CALL LCFIT(XD,YR,NXORIG,0,XNX,YDNX,1,0,CLSP)
      CALL LCFIT(XD,YCB,NXORIG,0,XNX,YCBNX,1,0,C(SPCB)
      A2=YDNX**EXP-YCBNX**EXP
      IF(A2.LE. A1)  STATE=.TRUE.
```

```
1001 CALL DTEST                                                JETPRP01
     RETURN                                                    JETPRP02
     END                                                       JETPRP03
```

```
CJMESHM          MESH REDISTRIBUTION FOR TRANS. REGION
      SUBROUTINE JMESHM
      COMMON /PARAM/ DUM(1601),SM(200),NM,DUM1(7)
      COMMON /CTRL/ DUM2(15), NPD, DUM3(80)
      COMMON /UMESH/ MCHANG,CK,DY1,NMSH      , CXPC,CXTP,NRED
C*
C*
    1 PSIE=SM(NPD)
      DPSIF=DY1*PSIE
      CK1=CK-1.
      CON=DPSIF/CK1
      DO 5 L=1,NMSH
      EXP=FLOAT(L-1)
    5 SM(L)=CON*(CK**EXP-1.)
      NPD=NMSH
   10 RETURN
      END
```

```
CJTCTRL          MAIN JET CONTROL ROUTINE
      SUBROUTINE JTCTRL
      LOGICAL SUPC,SUPSTP
      LOGICAL MCHANG
      LOGICAL CURSTP
      LOGICAL BYPASS
      LOGICAL  EOF   ,  ERR
      LOGICAL  AXI , XPRN , CMPRS , QJET , TURBJ , CORE
      REAL  MJET , ME , MUREF
      COMMON /RSTART/ NREG,RESTRT,NRES,MIXPRE
      LOGICAL  MIXPRE
C*
      COMMON /JETTWO/
     * TWO     , DIAO    , MJETO   , TJETO   , VJETO   ,
     * PTJETO , TIJETO , NJO
      REAL  MJETO,MACHO
      COMMON /BCO/  UO, EO, THO
      COMMON /CTRL2/
     * EDGEI   , SFI     , MERGE   , XMERGE , YMERGE ,
     * SLOPEI , SLOPEO , CEPTI   , CEPTO
      COMMON /MERGET/  MER, MERSTP , XMRG
      LOGICAL  TWO, MERGE , MER , MERSTP
C****** INPUT COMMON
C*
      COMMON /INPJET/
     * DIAJ          , MJET          , TJET          , PTJET         , VJET          ,
     * TIJET         ,
     * PE            , VE            , ME            , TIE           , TE            ,
     * AXI           , NJ            , NMAX          ,
     * X(100)        , XPRN(100)     ,
     * GAM           , RG            , PR            , PRT           ,
     * SC            , TREF          , MUREF
C*
C****** CONTROL COMMON
C*
      COMMON /CTRL/
     * NXTA          , CMPRS         , QJET          , TURBJ         , COEF(10)      ,
     * NPU           , NPD           , DXC           , XU            , XD            ,
     * DSTOR(800)
C*
C****** PROFILE COMMON
C*
      COMMON /PROF/ PSI(200),Y(200),UD(200),THD(200),ED(200)
C*
C****** CONSTANT AND ERROR COMMON
C*
      COMMON /CNERR/ BITS , ERR , GC , GCJ , FOOT
C*
C****** BOUNDARY CONDITION COMMON
C*
```

```
      COMMON /BC/ UEDGE  , EEDGE  , THEDGE
C*
C****** POTENTIAL CORE COMMON
C*
      COMMON /CORED/ XCORE  , CORE  , CORSTP
      COMMON /SUPER/ SUPC,SUPSTP,XSUP
C*
C****** SCALER (UNITS CONVERSION) COMMON
C*
      COMMON /SCALER/  SP  , SV  , SLEN
C*
C****** JET PROPERTIES COMMON
C*
      COMMON /JET/
     * B(100)   , UC(100)  , TC(100)  , TIC(100) ,
     * PTC(100) , WJ(100)  , YJ(100)  ,
     * YSONIC(100)
      COMMON /JET1/ FLOWJ, TTO, NXN , EJET
      COMMON /JET2/  TTC(100)
      COMMON /UMESH/ MCHANG,CK,DY1,NMSH      , CXPC,CXTP,NRED
      COMMON /JET3/ STADD, NV, STATF
      LOGICAL  STADD,STATF
      COMMON /MIXER/ MIX,RD(100),ZD(100),CF,YR(100)
      LOGICAL  MIX
      DATA BYPASS/F/
C*
C*  INITIALIZE AT FIRST STATION--INCLUDE STATION WHERE CORE
C*  DISAPPEARS AS A CALCULATION STATION TO BE INSERTED
C*
      NCALC=NXTA
    1 CORE=.FALSE.
      SUPC=.FALSE.
      MER=.FALSE.
      NXP=NRES
      NXN=NXP+1
      XU=X(NXP)
      IF(.NOT. MIX)  GO TO 2
      NV=2
      STADD=.FALSE.
      STATF=.FALSE.
    2 XD=X(NXN)
      DXC=XD-XU
C*
C*  CALL JTSTEP FOR INTEGRATION TO NEXT CALCULATION STATION
C*
    3 CALL JTSTEP
      IF(ERR) GO TO 1000
C*
C*  TEST FOR END OF POTENTIAL CORE
C*
```

```
C*
C*  ALSO TEST FOR DISAPPEARANCE OF SUPERSONIC CORE IF JET IS SUPERSONIC
C*
        IF(CORE .AND. CORSTP) GO TO 5
        IF(SUPC .AND. SUPSTP) GO TO 5
        IF(MER .AND. MERSTP)    GO TO 5
        GO TO 20
C*
C*  RELOCATE JET PROPERTIES UP 1 LOCATION--
C*  INSERT XDCORE AND SET XPRN T
C*
      5 ASSIGN 9 TO LGO
        XTEST = X(NXN)
        IF( (XTEST.NE.XCORE).AND.(XTEST.NE.XSUP).AND.(XTEST.NE.XMRG) )
       * GO TO 501
        ASSIGN 15 TO LGO
        GO TO 8
    501 NM     = -(NXTA-NXN+1)
        N1=NXN+1
        CALL MOVE(5,X(NXN),X(N1),NM,1,B(NXN),B(N1),NM,1,
       * TTC(NXN),TTC(N1),NM,1,XPRN(NXN),XPRN(N1),NM,1,
       * YSONIC(NXN),YSONIC(N1),NM,1)
        CALL MOVE(5,YJ(NXN),YJ(N1),NM,1,UC(NXN),UC(N1),NM,1,
       * TC(NXN),TC(N1),NM,1,PTC(NXN),PTC(N1),NM,1,
       * TIC(NXN),TIC(N1),NM,1)
        CALL MOVE(1,WJ(NXN),WJ(N1),NM,1)
      6 IF(CORE) X(NXN)=XCORE
        IF(SUPC) X(NXN)=XSUP
        IF(MER)  X(NXN)=XMRG
      7 XPRN(NXN)=.TRUE.
      8 IF( CORE ) CORE=.FALSE.
        IF(SUPC) SUPC=.FALSE.
        IF(MER)  MER=.FALSE.
        GO TO LGO , (9,15)
      9 NCALC = NCALC+1
        NXTA=NXTA+1
C*
C*  COMPUTE JET PROPERTIES
C*
     15 IF(MIX) STADD=.TRUE.
     20 CALL JETPRP
     21 IF(ERR) GO TO 1000
C*
C*  PRINT PROFILES IF REQUESTED
C*
     30 IF(XPRN(NXN))  CALL JTOUTP
C*
C*  IF CORSTP=T, REDISTRIBUTE MESH AFTER DISAPPEARANCE
C*  OF POTENTIAL CORE
C*
```

```
   40 IF(.NOT. CORSTP) GO TO 50
      IF(BYPASS) GO TO 50
      IF(MCHANG) CALL JMESHM
      BYPASS=.TRUE.
C*
C*  INCREMENT COUNTERS,,,   TEST FOR END OF PROBLEM
C*
   50 NXN=NXN+1
      NXP=NXP+1
      IF(NXN.GT. NCALC) GO TO 1000
      IF(MIX .AND. STATF)  GO TO 1000
      IF(.NOT. STADD)  NV=NV+1
      IF(MIX)  STADD=.FALSE.
C*
C*  CONTINUE INTEGRATION
C*
  100 XU=XD
      GO TO 2
C*
 1000 RETURN
      END
```

```
CJTEDGE         LOCATE EDGE OF JET        NEW Y CO-ORDINATES
      SUBROUTINE JTEDGE(X,YE,PSIE,ADDP)
      LOGICAL HALF
      LOGICAL AXI
      COMMON /JETTWO/
     * TWO    , DIAO   , MJETO  , TJETO  , VJETO  ,
     * PTJETO , TIJETO , NJO
      REAL  MJETO,MACHO
      COMMON /BCO/  UO, EO, THO
      COMMON /CTRL2/
     * EDGEI  , SFI    , MERGE  , XMERGE , YMERGE ,
     * SLOPEI , SLOPEO , CEPTI  , CEPTO
      COMMON /MERGET/  MER, MERSTP , XMRG
      COMMON /PROPJ2/  MACHO,REFLO,YII,YO,MERGP
      LOGICAL MERGP
      LOGICAL  TWO, MERGE , MER , MERSTP
      COMMON /MISC/ PM(10)
      LOGICAL ADDP
      COMMON /SETNEW/  LKK,LCOR
      COMMON /PROF/ PSI(200),Y(200),UD(200),THD(200),ED(200)
      COMMON /YOFXI/ LEI
      COMMON /INPJET/ DIAJ,MJET,TJET,DUMI1(7),TE,AXI,DUMI2(203),RG
      COMMON /BC/ UEDGE,EEDGE,THEDGE
      COMMON /PROPJ1/ P,DUMPR(1412)
      COMMON /PARAM/ UDIF(200),PSI1(200),DUMP(1201),SM(200),NM,
     * DUMP1(7)
      COMMON /ERASE/  DUME(800)
      COMMON /CTRL/ DUMC1(15), NPD  , DUMC2(803)
      COMMON /XPRIN/ DPRIN
C*
C*  AITER1-- COMPUTE B.C. FOR NEXT TRIAL AT MASS BALANCE
C*
      COMMON /MIXER/ MIX,RD(100),XD(100),CF,YR(100)
      LOGICAL  MIX
      COMMON /FLOBAL/ MAXIT,SUPB,NIT,PSID,YDD,YDC,
     *                P1,P2,UCL,TOL,UPSTRM,CVG
      LOGICAL  SUPB,CVG,UPSTRM
      COMMON /CNERR/ BITS,ERR,GC,GCJ,FOOT
      LOGICAL ERR
      COMMON /ACONVG/ YCD(100),PD(100),INDC(100),  CHOKE, CHOKED
      LOGICAL  CHOKE, CHOKED
      COMMON /DFIT/  CLSP(100)
      COMMON /STA2/ MACH2,TS2,SS2,V2,RHO2,DPDX2
      REAL  MACH2
      COMMON /BCMIX2/ GRADU,TW,MUW,RHOW,PTF,TTE
      REAL  MUW
      COMMON /CBODY/  YCB(100),CLSPCB(100),YCB1 , UCL1
      COMMON /DUIMIX/ NXORIG
C*
      LOGICAL DPRIN
```

```
      COMMON /SCALED/ SCLD,ALXLIM
      LOGICAL         SCLD
      COMMON /MOLES / ALX(100,12)
      COMMON /DIFEQI/ NC,DDIF(120)
      EQUIVALENCE (IE, LE1 )
      DIMENSION STORI(200)
      EQUIVALENCE (STORI(1),DUMP(1))
      DIMENSION YI(200),RHO(200)
      EQUIVALENCE (YI(1),DUMP(201)),(  RHO(1),DUMP(401))
      DATA LFFF/200/
      DATA IOF/0/
C*
C*
C*  IF HALF=.TRUE. EDGE IS TAKEN AT 50 PCT VELOCITY LINE  (Y102)
*
      IE    = 1
      ADDP=.FALSE.
      HALF=.FALSE.
      IF(PM(4).NE.0.) HALF=.TRUE.
C*
C*  CHECK JET UD FOR 1-ST POINT WHERE UD=UEDGE
C*  REDEFINE JET EDGE CONDITIONS
      DO 6620 L=1,NPD
      IF(UD(L).EQ.UEDGE  .AND. UEDGE.EQ. 0.)  GO TO 6621
 6620 CONTINUE
 6621 NPD=L
      IF((.NOT. MIX)  .OR.  (MIX .AND. UPSTRM))  THD(NPD)=THEDGE
      IF((.NOT. MIX)  .OR.  (MIX .AND. UPSTRM))   ED(NPD)=EEDGE
      NPD1=NPD-1
      DIFEU=ABS(UEDGE-UD(NPD-1))
      IF(.NOT. MIX) GO TO 1
      DO 6720 L=1,NPD
 6720 RHO(L)=144.*P2/(RG*THD(L)*TJET)
      LE1=NPD
      LF=LE1-1
      PSTE=SM(LE1)
      RHOE=RHO(LE1)
      UEE=UD(LE1)
      GO TO 468
C*
C*
C*  LOCATE EDGE(S) OF JET(S)
C*  -- POINT WHERE VELOCITY DIFFERENCE IS  .98*MAX. VELOCITY
C*  DIFFERENCE ACROSS THE JET
C*
C*
    1 UCL1=UD(1)
      LK=1
      DO 2 L=1,NPD
      UDIF(L)=ABS(UD(L)-UCL1)
```

```
      IF(L.EQ.1)  GO TO 2                                                JTEDGE01
      TESTA=UD(L-1)+1.E-6                                                JTEDGE02
      TESTB=UD(L-1)-1.E-6                                                JTEDGE03
      IF(UD(L).GE.TESTB .AND. UD(L).LE.TESTA .AND. L.LT.LFFF) LK=L       JTEDGE04
    2 RHO(L)=144.*P/(RG*THD(L)*TJET)                                     JTEDGE05
      TESTE=AMAX1(.98,PM(3))                                             JTEDGE06
C                                                                        JTEDGE07
C     DETERMINE EDGE FROM MOLE FRACTION                                  JTEDGE08
C                                                                        JTEDGE09
      IF( IOF.EQ.0 ) GO TO 2220                                          JTEDGE10
      IF( .NOT.SCLD ) GO TO 2220                                         JTEDGE11
      TESTE = AMAX1( .98,PM(3) )                                         JTEDGE12
      DO 2208 LE1=1,NPD                                                  JTEDGE13
      IF( ALX(LE1,NC).GE.TESTE ) GO TO 2209                              JTEDGE14
 2208 CONTINUE                                                           JTEDGE15
 2209 LE     = LE1-1                                                     JTEDGE16
      SLOPED= (SM(LE1)-SM(LE))/(ALX(LE1,NC)-ALX(LE,NC))                  JTEDGE17
      PSIE   = SM(LE)+SLOPED*(TESTE-ALX(LE,NC))                          JTEDGE18
      CALL MOVE(1,SM,PSI1,LE,1)                                          JTEDGE19
      PSI1(LE1)= PSIE                                                    JTEDGE20
      DO 2210 L=1,LE                                                     JTEDGE21
 2210 STORI(L)= 2./(RHO(L)*UD(L))                                        JTEDGE22
      RHOE   = YOF(PSIE,SM,RHO,1,NPD)                                    JTEDGE23
      UEE    = YOF(PSIE,SM,UD,1,NPD)                                     JTEDGE24
      STORI(LE1)= 2./(RHOE*UEE)                                          JTEDGE25
      YI(1) = 0.                                                         JTEDGE26
      CALL INTG(STORI,PSI1,YI,2,LE1)                                     JTEDGE27
      YE     = YI(LE1)                                                   JTEDGE28
      YSV    = YE                                                        JTEDGE29
      IF( AXI ) YE=SQRT(2.*YE)                                           JTEDGE30
      CALL MOVE(1,YI,Y,LE,1)                                             JTEDGE31
      CALL MOVE(1,ALX(1,NC),UDIF,NPD,1)                                  JTEDGE32
      GO TO 4001                                                         JTEDGE33
 2220 ULIM   = TESTE*ABS(UEDGE-UCL1)                                     JTEDGE34
      IF(HALF) ULIM=.5*ABS(UEDGE-UD(1))                                  JTEDGE35
      LFFF=LK                                                            JTEDGE36
      IF(DPRIN)  CALL TABPRT(2HLK,LK,1,1,0)                              JTEDGE37
      IF(DPRIN) CALL TABPRT(4HUDIF,UDIF,NPD,10,0)                        JTEDGE38
C*                                                                       JTEDGE39
C*                                                                       JTEDGE40
C****      LOCATE OUTER STREAMLINE     * ********                        JTEDGE41
C*                                                                       JTEDGE42
C*  FOR OUTER STREAMLINE-- CHOOSE THE MINIMUM OF INTERPOLATED            JTEDGE43
C*      AND EXTRAPOLATED VALUES                                          JTEDGE44
C*                                                                       JTEDGE45
      LE1=0                                                              JTEDGE46
      L      = NPD+1                                                     JTEDGE47
  300 L      = L-1                                                       JTEDGE48
      IF( UDIF(L).LT.ULIM ) GO TO 301                                    JTEDGE49
      IF( L.GE.LK ) GO TO 300                                            JTEDGE50
```

```
  301 LE1   = L+1                                                   JTEDGE51
      LF=LE1-1                                                      JTEDGE52
      SLOPED=(SM(LE1)-SM(LE))/(UDIF(LE1)-UDIF(LE))                  JTEDGE53
      PSIE1=SM(LF)+SLOPED*(ULIM-UDIF(LE))                           JTEDGE54
      IF(.NOT. HALF) GO TO 6                                        JTEDGE55
      PSIE=PSIE1                                                    JTEDGE56
      LEE   = LE                                                    JTEDGE57
      GO TO 8                                                       JTEDGE58
    6 LEE=LE1-2                                                     JTEDGE59
      SLOPEE=(SM(LE)-SM(LEE))/(UDIF(LE)-UDIF(LEE))                  JTEDGE60
      PSIE2=SM(LF)+SLOPEE*(ULIM-UDIF(LE))                           JTEDGE61
      PSIE=AMIN1(PSIE1,PSIE2)                                       JTEDGE62
    8 UEE   = YOF(PSIE,SM,UD,LEE,NPD)                               JTEDGE63
      RHOE=YOF(PSIE,SM,RHO,LJ,NPD)                                  JTEDGE64
  468 DO 4 L=1,LF                                                   JTEDGE65
    4 STORI(L)=2./(RHO(L)*UD(L))                                    JTEDGE66
      STORI(LE1)=2./(RHOE*UEE)                                      JTEDGE67
C*                                                                  JTEDGE68
C*                                                                  JTEDGE69
C*   MOVE PSI ARRAY TO SCRATCH--STORAGE                             JTEDGE70
C*                                                                  JTEDGE71
      CALL MOVE(1,SM,PSI1,LE,1)                                     JTEDGE72
C*                                                                  JTEDGE73
C*   INSERT EDGE STREAMLINE                                         JTEDGE74
C*                                                                  JTEDGE75
      PSI1(LE1)=PSIE                                                JTEDGE76
C*                                                                  JTEDGE77
C*   DETERMINE Y- COORDINATES BY INTEGRATION OF STREAM FUNCTION     JTEDGE78
C*                                                                  JTEDGE79
C*                                                                  JTEDGE80
C*   USE TRAPEZOIDAL RULE                                           JTEDGE81
C*                                                                  JTEDGE82
      YI(1)=0.                                                      JTEDGE83
      YCB1=0.                                                       JTEDGE84
      IF(.NOT. MIX)  GO TO 399                                      JTEDGE85
      CALL LCFIT(XD,YCB,NXORIG,0,X*DIAJ,YI(1),1,0,CLSPCB)           JTEDGE86
      YCB1=YI(1)                                                    JTEDGE87
      IF(AXI) YI(1)=0.                                              JTEDGE88
  399 CALL INTG(STORI,PSI1,YI,2,LE1)                                JTEDGE89
      YE=YI(LE1)                                                    JTEDGE90
      YSV=YE                                                        JTEDGE91
      IF(AXI) YE=SQRT(YCB1**2+2.*YE)                                JTEDGE92
C*   MOVE YI VALUES TO Y ARRAY                                      JTEDGE93
C*                                                                  JTEDGE94
      LX=LF                                                         JTEDGE95
      IF(MIX) LX=LE1                                                JTEDGE96
   40 CALL MOVE(1,YI,Y,LX,1)                                        JTEDGE97
      IF(MIX) GO TO 8888                                            JTEDGE98
C*                                                                  JTEDGE99
C*   CONTINUE INTEGRATION TO NPD-1                                  JTEDGE00
```

```
C*  EXTRAPOLATE FOR NPD POINT
C*
 4001 NPD1  = NPD-1
      MX=1
      IF(NPD1.EQ. LE)  MX=2
      IF(NPD1.EQ. LE)  GO TO  887
      DO 42 L=LE1,NPD1
   42 STORI(L)=2./(RHO(L)*UD(L))
      CALL INTG(STORI,SM,Y,LE1,NPD1)
  887 IF(UEDGE .GT. 0.)  GO TO 8887
      GO TO (888,889), MX
C 888 SLOPE=(Y(NPD1)-Y(NPD1-1))/(UDIF(NPD1)-UDIF(NPD1-1))
C     Y(NPD)=Y(NPD1)+SLOPE*(UDIF(NPD)-UDIF(NPD1))
  888 SLOPE=(Y(NPD1)-Y(NPD1-1))/(SM(NPD1)-SM(NPD1-1))
      Y(NPD)=Y(NPD1)+SLOPE*(SM(NPD)-SM(NPD1))
      GO TO  8888
C 889 SLOPE=(YSV-Y(NPD1))/(ULIM-UDIF(NPD1))
C     Y(NPD)=YSV+SLOPE*(UDIF(NPD)-ULIM)
  889 SLOPE=(YSV-Y(NPD1))/(PSIE-SM(NPD1))
      Y(NPD)=YSV+SLOPE*(SM(NPD)-PSIE)
      GO TO 8888
 8887 STORI(NPD)=2./(RHO(NPD)*UD(NPD))
      CALL INTG(STORI,SM,Y,NPD,NPD)
 8888 Y(1)=YCB1
      IF(.NOT. AXI)  GO TO 3811
      IF(.NOT.DPRIN) GO TO 19
      CALL TABPRT(4HPSI1,PSI1,LE,10,0)
      CALL TABPRT(4HY(L),Y,NPD,10,0)
      CALL TABPRT(5HYI(L),YI,LE,10,0)
      CALL TABPRT(5HSTORI,STORI,LE,10,0)
      WRITE(6,107)PM(4),UEDGE,ULIM,PSIE,NPD,NPD1,LE1,LE,MX,HALF,MIX,YCB1
  107 FORMAT(1X//
     1 2X,8HPM(4)    =,E14.6,2X,8HUEDGE    =,E14.6,2X,8HULIM     =,E14.6,2X,
     2 8HPSIE    =,E14.6//
     3 2X,8HNPD      =,I6,10X,8HNPD1     =,I6,10X,8HLE1      =,I6,10X,
     4 8HLF      =,I6//
     5 2X,8HMX       =,I6,10X,8HHALF     =,L6,10X,8HMIX      =,L6,10X,
     6 8HYCB1    =,E14.6)
   19 CONTINUE
      DO 7 L=2,NPD
    7 Y(L)=SQRT(YCB1**2+2.*Y(L))
 3811 CONTINUE
C*
C*  DETERMINE IF NEW POINT IS TO BE ADDED
C*
C*
C*  IF TWO=T  AND MERGE=F,  LOCATE BOUNDARY OF INNER JET
C*
      IF(MERGE .OR.  (.NOT. TWO)) GO TO 30
      ULIML=TESTE*ABS(U0-1.)
```

```
      TEST2=U0+1.E-6
      TEST1=U0-1.E-6
C*
C*  LOCATE NEAREST MESH POINT
C*
      DO 200 L=1,NPD
      IF(UD(L).LE.TEST2  .AND.  UD(L).GE. TEST1)  UD(L)=U0
      IF(UD(L) .EQ.  U0)  GO TO 259
  200 CONTINUE
C*
C*  ASSUME JETS HAVE MERGED-- SET CO-ORDINATES OF MERGE POINT
C*
      MERGE=.TRUE.
      MERGP=.TRUE.
      SFI=SM(NMERG)
      EDGEI=Y(NMERG)
      XMERGE=X
      YMERGE=EDGEI
C*
C*  COMPUTE COEFFICIENTS IN LINEAR EQUATIONS CONNECTING THE
C*  NOZZLE CORNERS WITH MERGE POINT
C*
      CEPTI=DIAO/DIAJ
      SLOPEI=(YMERGE-CEPTI)/XMERGE
      CEPTO=1.
      SLOPEO=(YMERGE-1.)/XMERGE
      GO TO 30
C*
C*  SET NMERGE TO L+1, STORE INNER EDGE SF AND CO-ORDINATE
C*
  259 NU=L
      DO 260 L=1,NU
      IF(UDIF(L).GE.ULIML)  GO TO 210
  260 CONTINUE
  210 NMERG=L
      NMRG1=L-1
      SLOPS=(SM(NMERG)-SM(NMRG1))/(UDIF(NMERG)-UDIF(NMRG1))
      SFI=SM(NMRG1)+SLOPS*(ULIML-UDIF(NMRG1))
      SLOPY=(Y(NMERG)-Y(NMRG1))/(SM(NMERG)-SM(NMRG1))
      EDGEI=Y(NMRG1)+SLOPY*(SFI-SM(NMRG1))
   30 TESTU=.005
      IF(PM(2).NE. 0.) TESTU=PM(2)
      IF(LKK.LE. 3  .AND.  UDIF(NPD).EQ.UDIF(NPD1))  LKK=4
      IF(DIFFU.GT. TESTU .OR.  LKK.LE.3 ) ADDP=.TRUE.
      LKK=LKK+1
      IF(DPRIN)  WRITE (6,7222) ADDP,DIFFU,TESTU
 7222 FORMAT(//12X,L6,2X,2E16.8)
C*
  100 RETURN
      END
```

```
CJTFILE        STORAGE OF AERO-DATA FILE FOR PLOT OR NOISE                JTFILE01
      SUBROUTINE JTFILE( NTRY,XX )                                        JTFILE02
      LOGICAL  ENTRYI                                                     JTFILE03
      LOGICAL FOUND                                                       JTFILE04
      LOGICAL MCHANG                                                      JTFILE05
      LOGICAL AMBTO                                                       JTFILE06
      LOGICAL OPRIN                                                       JTFILE07
      LOGICAL  EOF  ,  ERR                                                JTFILE08
      LOGICAL  AXI , XPRN , CMPRS , QJET , TURBJ , CORE                   JTFILE09
      LOGICAL ENTRY1                                                      JTFILE10
      REAL  MJET , ME , MUREF                                             JTFILE11
      REAL MACH                                                           JTFILE12
      REAL MOLF1(100),MOLF2(100),MOLF3(100),MOLF4(100),                   JTFILE13
     * MOLF5(100),MOLF6(100),MOLF7(100),MOLF8(100),                       JTFILE14
     * MOLF9(100),MOLF10(100),MOLF11(100),MOLF12(100)                     JTFILE15
      COMMON /CNFILE/ NF                                                  JTFILE16
      COMMON /RSTART/ NREG,RESTRT,NRES,MIXPRE                             JTFILE17
      LOGICAL  MIXPRE                                                     JTFILE18
      COMMON /CTRL2/ TWODT(9)                                             JTFILE19
      COMMON /DIFEQI/                                                     JTFILE20
     * NC,CNAME(12),ALJ(12),ALJO(12),ALE(12),                             JTFILE21
     * SCM(12),TCPRF(12),HCPRF(12),CPC(3,12)                              JTFILE22
      COMMON /DICTRL/ DIFF , CND(10)                                      JTFILE23
      LOGICAL          DIFF                                               JTFILE24
      COMMON /MOLES / ALX(1200)                                           JTFILE25
      COMMON /BCMOL / ALEDGE(12),ALO(12)                                  JTFILE26
      COMMON /FILK/  CSC                                                  JTFILE27
      COMMON /PARAM/  ,                                                   JTFILE28
     * U(200),T(200),TOT(200),XMACH(200),PTOT(200),TTD(200),              JTFILE29
     * PTD(200),DUMP9(409)                                                JTFILE30
      COMMON /RATIO/   AMBTO                                              JTFILE31
      COMMON /MISC/ PM(10),  PLOT                                         JTFILE32
      COMMON /INP1/ ENTRY1                                                JTFILE33
      COMMON /UMESH/ MCHANG,CK,DY1,NMSH      , CXPC,CXTP,NRED             JTFILE34
C*                                                                        JTFILE35
C****** INPUT COMMON                                                      JTFILE36
C*                                                                        JTFILE37
      COMMON /INPJET/                                                     JTFILE38
     * DIAJ      , MJET      , TJET      , PTJET     , VJET      ,        JTFILE39
     * TIJET     ,                                                        JTFILE40
     * PE        , VE        , ME        , TIE       , TE        ,        JTFILE41
     * AXI       , NJ        , NM        ,                                JTFILE42
     * X(100)    , XPRN(100) ,                                            JTFILE43
     * GAM       , RG        , PR        , PRT       ,                    JTFILE44
     * SC        , TREF      , MUREF                                      JTFILE45
C*                                                                        JTFILE46
C****** CONSTANT AND ERROR COMMON                                         JTFILE47
C*                                                                        JTFILE48
      COMMON /CNERR/ BITS , ERR , GC , GCJ , FOOT                         JTFILE49
C*                                                                        JTFILE50
```

```
C****** BOUNDARY CONDITION COMMON                                       JTFILE51
C*                                                                      JTFILE52
      COMMON /BC/ UEDGE  , EEDGE  , THEDGE                              JTFILE53
C*                                                                      JTFILE54
C****** POTENTIAL CORE COMMON                                           JTFILE55
C*                                                                      JTFILE56
      COMMON /CORED/ XCORE  , CORE  , CORSTP                            JTFILE57
C*                                                                      JTFILE58
C****** SCALER (UNITS CONVERSION) COMMON                                JTFILE59
C*                                                                      JTFILE60
      COMMON /SCALER/  SP  , SV  , SLEN                                 JTFILE61
C*                                                                      JTFILE62
      COMMON /JET1/ FLOWJ,TT0,NX,EJET                                   JTFILE63
      COMMON /PROPJT/                                                   JTFILE64
     * P        , PRL      , PRTT     , RGAS     , SCC      ,           JTFILE65
     * TREF     , VSREF    , MACH     , XLC      ,                      JTFILE66
     * REFL,C,CHI,RNORM,                                                JTFILE67
     * RHO(200),DUM13(600),XLN(200),DUM14(400)                          JTFILE68
      COMMON /XPRIN/ DPRIN                                              JTFILE69
      COMMON /FDATA/  DUM7(4),NFCSV,NRW                                 JTFILE70
      COMMON /CPROP/ CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8  ,CT9,CT10         JTFILE71
      COMMON/ PROF/ PSI(200),Y(200),UD(200),THD(200),ED(200)            JTFILE72
      COMMON /JET/                                                      JTFILE73
     * B(100),UC(100),TC(100),TIC(100),                                 JTFILE74
     * PTC(100),WJ(100),YJ(100),YSONIC(100)                             JTFILE75
      COMMON /JET2/ TTC(100)                                            JTFILE76
      COMMON /CTRL/                                                     JTFILE77
     * NXTA        , CMPRS       , QJET        , TURBJ       , COEF(10)    , JTFILE78
     * NPU  ,  NPD  , DXC  , XU  , XDD  ,                               JTFILE79
     * TTD(200)     , DSTOR(600)                                        JTFILE80
      COMMON /TAG/                                                      JTFILE81
     * NAME(10),TITLE1(10),IDENT(10),ADDRES(10),IDENT1(10)              JTFILE82
C*                                                                      JTFILE83
      COMMON /MIXER/ MIX,RD(100),XD(100),CF,YR(100)                     JTFILE84
      LOGICAL  MIX                                                      JTFILE85
      COMMON /FLOBAL/ MAXIT,SUPB,NIT,PSID,YDD,YDC,                      JTFILE86
     *                P1,P2,UCL,TOL,UPSTRM,CVG                          JTFILE87
      LOGICAL  SUPB,CVG,UPSTRM                                          JTFILE88
      COMMON /ACONVG/ YCD(100),PD(100),INDC(100),  CHOKE, CHOKED        JTFILE89
      LOGICAL  CHOKE, CHOKED                                            JTFILE90
      COMMON /DFT1/  CLSP(100)                                          JTFILE91
      COMMON /STA2/ MACH2,TS2,SS2,V2,RHO2,DPDX2                         JTFILE92
      REAL  MACH2                                                       JTFILE93
      COMMON /BCMIX2/ GRADJ,TW,MUW,RHOW,PTE,TTE                         JTFILE94
      REAL  MUW                                                         JTFILE95
      COMMON /MIXPRP/ MA2(100),VE2(100),TE2(100),IWC(100)               JTFILE96
      REAL MA2                                                          JTFILE97
      COMMON /THRST/ WV(100)                                            JTFILE98
      COMMON /OUTMIX/ NXORIG                                            JTFILE99
      COMMON /CBODY/  YCB(100),CLSPCB(100),YCB1 , UCL1                  JTFILE00
```

```
      COMMON /JET3/STADD,NV,STATF
      COMMON /KEYS/ KEY(11),KEYB(11),KODA(11),KODB(11)
      COMMON /MERGE1/ MER,MERSTP,XMRG
      COMMON /SUPER/ SUPC,SUPSTP,XSUP
      COMMON /CPROP2/ CTP,CTS,CTM
      COMMON /PROPJ2/ MACHO,REFLO,YI,YO,MERGE
      COMMON /CONSTF/ CON1
      COMMON /FILIND/ KREC,KXX
C*
      EQUIVALENCE (MOLF1,ALX(1)),(MOLF2,ALX(101)),(MOLF3,ALX(201)),
     *            (MOLF4,ALX(301)),(MOLF5,ALX(401)),
     *            (MOLF6,ALX(501)),(MOLF7,ALX(601)),
     *            (MOLF8,ALX(701)),(MOLF9,ALX(801)),
     *            (MOLF10,ALX(901)),(MOLF11,ALX(1001)),
     *            (MOLF12,ALX(1101))
C*
      DIMENSION GJET(200)
      DIMENSION  LEN2(11),NS2A(10),LEN1(5),NSA(4)
      EQUIVALENCE  (UE,VE)
      EQUIVALENCE (C6,COEF(6))
      DATA  ENTRYI/T/
      DATA NSA/15,1HJ,6,1HJ/
      DATA NS2A/5,2HII,1,2HIL,11,1HI,10,2HIJ,5,2HJK/
      DATA IBLANK/0202020202020/
      DATA IPROF/5HAPROF/
      DATA KXX1/2HXX/
C*
C*
C*    NTRY=1      WRITE PROFILES AT STATION XX
C*    NTRY=2    WRITE CL PROPS. AND INPUT VARIABLES
C*     NTRY=3    READ CL PROPS AND INPUT VARIABLES
C*    NTRY=4    RECLAIM PROFILE AT STATION XX
C*
    1 GO TO  (10,100,120,200) , NTRY
C*
   10 IF(.NOT.ENTRYI)  GO TO 11
      CALL FMPYC(1,VJET,UD,U,NPD)
      CALL FMPYC(1,TJET,THD,T,NPD)
      KREC  = 0
   11 ENTRYI=.FALSE.
      KREC  = KREC+1
      KXX   = CSC*XX+.5
      WRITE (3) KREC,KXX,KREG,
     * SUPD,SUPSTP,CORF,CORSTP,MER,MERSTP,NPD,
     * PSI,Y,UD,THD,ED,TID,RHO,XLN,U,T,TOT,XMACH,
     * PTOT,TID,PTD,
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,J
      GO TO 1000
C
```

```
  100 REWIND 3                                                          JTFILE51
      WRITE (2) KXX1,KREC,                                              JTFILE52
     * NAME,TITLE1,IDENT,ADDRES,IDENT1,TWODT,BITS,ERK,GC,GCJ,           JTFILE53
     * FOOT,DIAJ,MJET,TJET,PTJET,VJET,TIJET,EJET,PE,VE,ME,TIE,TE,       JTFILE54
     * AXI,NJ,NM,UE,MIXPRE,XLC,FLOWJ,MERGE,NV,CON1,                     JTFILE55
     * CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,CT9,CTP,CTS,CTM,                 JTFILE56
     * GAM,RG,PR,PRT,SC,TREF,MUREF,SP,SV,SLEN,DPRIN,PLOT,C6,            JTFILE57
     * MIX,CF,MAXIT,TOL,SUPB,                                           JTFILE58
     * X,XPRN,B,UC,TC,TIC,PTC,WJ,YJ,TTC,YSONIC,                         JTFILE59
     * YCB,XD,RD,YR,YCD,PD,WV,MA2,VE2,TE2,NXTA,I,                       JTFILE60
     * NC,CNAME,ALJ,ALJO,ALE,SCM,CPC,DIFF,CSC                           JTFILE61
      DO 101 IJ=1,KREC                                                  JTFILE62
      READ (3) JREC,KXX,KREG,                                           JTFILE63
     * SUPD,SUPSTP,CORE,CORSTP,MER,MERSTP,NPD,                          JTFILE64
     * PSI,Y,UD,THD,ED,TID,RHO,XLN,U,T,TOT,XMACH,                       JTFILE65
     * PTOT,TTD,PTD,                                                    JTFILE66
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,                             JTFILE67
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,J                         JTFILE68
      WRITE (2) JREC,KXX,KREG,                                          JTFILE69
     * SUPD,SUPSTP,CORE,CORSTP,MER,MERSTP,NPD,                          JTFILE70
     * PSI,Y,UD,THD,ED,TID,RHO,XLN,U,T,TOT,XMACH,                       JTFILE71
     * PTOT,TTD,PTD,                                                    JTFILE72
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,                             JTFILE73
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,J                         JTFILE74
  101 CONTINUE                                                          JTFILE75
      GO TO 1000                                                        JTFILE76
C                                                                       JTFILE77
  120 READ (1)                                                          JTFILE78
     * DIAJ,MJET,TIJET,VJET,PE,TE,TIE,VE,GEX,RG,PR,PRT,SC,TREF,         JTFILE79
     * MUREF,DIFF,NC,CNAME,ALE,SCM,CPC,NJ,NM,CT1,CT2,CT3,CT4,CT5,       JTFILE80
     * CT6,CT7,CT8,GJET,Y,UD,THD,TID,ED,                                JTFILE81
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,                             JTFILE82
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12                           JTFILE83
      IF( NF.GE.2 ) READ (2) KXX1,KREC,                                 JTFILE84
     * NAME,TITLE1,IDENT,ADDRES,IDENT1,TWODT,BITS,ERK,GC,GCJ,           JTFILE85
     * FOOT,DIAJ,MJET,TJET,PTJET,VJET,TIJET,EJET,PE,VE,ME,TIE,TE,       JTFILE86
     * AXI,NJ,NM,UE,MIXPRE,XLC,FLOWJ,MERGE,NV,CON1,                     JTFILE87
     * CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,CT9,CTP,CTS,CTM,                 JTFILE88
     * GAM,RG,PR,PRT,SC,TREF,MUREF,SP,SV,SLEN,DPRIN,PLOT,C6,            JTFILE89
     * MIX,CF,MAXIT,TOL,SUPB,                                           JTFILE90
     * X,XPRN,B,UC,TC,TIC,PTC,WJ,YJ,TTC,YSONIC,                         JTFILE91
     * X,XPRN,B,UC,TIC,PTC,WJ,YJ,TTC,YSONIC,                            JTFILE92
     * YCB,XD,RD,YR,YCD,PD,WV,MA2,VE2,TE2,NXTA,I,                       JTFILE93
     * NC,CNAME,ALJ,ALJO,ALE,SCM,CPC,DIFF,CSC                           JTFILE94
      GO TO 1000                                                        JTFILE95
C                                                                       JTFILE96
  200 KRES= CSC*XX+.5                                                   JTFILE97
      DO 210 IJ=1,KREC                                                  JTFILE98
      READ (2) JREC,KXX,KREG,                                           JTFILE99
     * SUPD,SUPSTP,CORE,CORSTP,MER,MERSTP,NPD,                          JTFILE00
```

```
     * PSI,Y,UD,THD,ED,TID,RHO,XLN,U,T,TOT,XMACH,                      JTFILE01
     * PTOT,TTD,PTD,                                                   JTFILE02
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,                            JTFILE03
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,J                        JTFILE04
       WRITE (3) JREC,KXX,KREG,                                        JTFILE05
     * SUPD,SUPSTP,CORE,CORSTP,MER,MERSTP,NPD,                         JTFILE06
     * PSI,Y,UD,THD,ED,TID,RHO,XLN,U,T,TOT,XMACH,                      JTFILE07
     * PTOT,TTD,PTD,                                                   JTFILE08
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,                            JTFILE09
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,J                        JTFILE10
       IF( KXX.EQ.KRES ) GO TO 220                                     JTFILE11
  210 CONTINUE                                                         JTFILE12
       ERR    = .TRUE.                                                 JTFILE13
       WRITE (6,211) XX                                                JTFILE14
  211 FORMAT(1H1,///2X,10HSTATION X=,F10.6,2X,11HNOT ON FILE//)        JTFILE15
       GO TO 1000                                                      JTFILE16
  220 WRITE (6,221) XX                                                 JTFILE17
  221 FORMAT(1H1,///2X,20H**JETMIX RESTART  X=,F10.6//)                JTFILE18
       IF(MIX) MIXPRE=.TRUE.                                           JTFILE19
C                                                                      JTFILE20
 1000 RETURN                                                           JTFILE21
       END                                                             JTFILE22
```

```
CJTINIT        INITIALIZE NEW PROBLEM
      SUBROUTINE JTINIT
C***********************************************************************
C********************** SPECIAL VERSION FOR 12 SPECIES *****************
C***********************************************************************
      LOGICAL DPRIN
      INTEGER  PLOT
      LOGICAL ENTRY1
      LOGICAL MCHANG
      LOGICAL ERR
      LOGICAL  AXI , XPRN , CMPRS , QJET , TURBJ , CORE
      REAL  MJET , ME , MUREF
      COMMON /RSTART/ NREG,RESTRT,NRES,MIXPRE
      LOGICAL  MIXPRE
      COMMON /DIFEQT/
     * NC , CNAME(12) , ALJ(12) , ALJO(12) ,ALE(12) , SCM(12) ,
     * TCPRF(12) , HCPRF(12) , CPC(3,12)
      COMMON /DICTRL/ DIFF , CND(10)
      LOGICAL        DIFF
      COMMON /MOLES / ALX(100,12)
      COMMON /BCMOL / ALEDGE(12),ALO(12)
C*
C****** INPUT COMMON
C*
      COMMON /INPJET/
     * DIAJ       , MJET        , TJET        , PIJET       , VJET       ,
     * TTJET      ,
     * PE         , VE          , ME          , TIE         , TE         ,
     * AXI        , NJ          , NM          ,
     * X(100)     , XPRN(100)   ,
     * GAM        , RG          , PR          , PRT         ,
     * SC         , TREF        , MUREF
C*
C*
C*  MESH COMMON AFTER DISAPPEARANCE OF POTENTIAL CORE
C*
C****** CONTROL COMMON
C*
      COMMON /CTRL/
     * NXTA        , CMPRS       , QJET        , TURBJ       , COEF(10)   ,
     * NPU  ,  NPD  , DXC  , XU  , XDD  ,
     * DSTOR(800)
C*
C****** PROFILE COMMON
C*
      COMMON /PROF/ PST(200),Y(200),  PRF(200,3)
C*
C****** CONSTANT AND ERROR COMMON
C*
      COMMON /CNERR/ BITS , ERR , GC , GCJ , FOOT
```

```
C*
C****** BOUNDARY CONDITION COMMON
C*
      COMMON /BC/ UEDGE  , EEDGE  , THEDGE
C*
C****** POTENTIAL CORE COMMON
C*
      COMMON /CORED/ XCORE  , CORE  , CORSTP
C*
C****** SCALER (UNITS CONVERSION) COMMON
C*
      COMMON /SCALER/  SP  , SV  , SLEN
C*
C****** JET PROPERTIES COMMON
C*
      COMMON /JET/
     * B(100)   , UC(100)   , TC(100)   , TIC(100) ,
     * PIC(100) , WJ(100)   , YJ(100)   ,
     * YSONIC(100)
      COMMON /JET1/ FLOWJ,ITO,NX,EJET
      COMMON /UMESH/ MCHANG,CK,DY1,NMSH      , CXPC,CXTP,NRED
      COMMON /EDGE/  YJETE  ,  SEEDGE
C*
      COMMON /MISC/ PM(10),  PLOT
      COMMON /INP1/  ENTRY1
      COMMON /JETTWO/
     * TWO    , DIAO   , MJETO  , TJETO  , VJETO  ,
     * PTJETO , TTJETO , NJO
      COMMON /BCO/  UO, EO, THO
      REAL  MJETO,MACHO
      COMMON /CTRL2/
     * EDGEI  , SEI    , MERGE  , XMERGE , YMERGE ,
     * SLOPEI , SLOPEO , CEPTI  , CEPTO
      COMMON /MERGET/  MER, MERSTP , XMRG
      LOGICAL  TWO, MERGE , MER , MERSTP
C*
      COMMON /MIXER/ MIX,RD(100),XD(100),CF,YR(100)
      LOGICAL  MIX
      COMMON /FLOBAL/ MAXIT,SUPB,NIT,PSID,YDD,YDC,
     *                P1,P2,UCL,IOL,UPSTRM,CVG
      LOGICAL  SUPB,CVG,UPSTRM
      COMMON /ACONVG/ YCD(100),PD(100),INDC(100),  CHOKE, CHOKED
      LOGICAL  CHOKE, CHOKED
      COMMON /DEIT/  CLSP(100)
      COMMON /STA2/ MACH2,TS2,SS2,V2,RHO2,DPDX2
      REAL  MACH2
      COMMON /BCMIX2/ GRADU,TW,MUW,RHOW,PTE,TTE
      REAL  MUW
      COMMON /CBODY/  YCB(100),CLSPCB(100),YCB1 , UCB1
      COMMON /OUTMIX/ NXORIG
```

```
      COMMON /INNAME/ NB,TAB(5),ND,TAD(4)                               JTINIT01
      COMMON /SCALED/ SCLD,ALXLIM                                       JTINIT02
      LOGICAL         SCLD                                              JTINIT03
      COMMON /CRMOD / CJRMOD                                            JTINIT04
      DIMENSION TAB1(5),TAD1(4)                                         JTINIT05
      DIMENSION TMP(10)                                                 JTINIT06
      EQUIVALENCE (TMP(1),PM(1))                                        JTINIT07
      DIMENSION  HH(3), CPA(3), CPCO2(3), CPH2O(3)                      JTINIT08
      DIMENSION CTABN(6)                                                JTINIT09
      DATA CTABN/3HAIR,3HCO2,3HH2O,6H      ,6H      ,6H      /          JTINIT10
      DATA HH/41523.74,-164431.94,-119593.66/                           JTINIT11
      DATA CPA/6.4303,8.929E-4,5.989F-8/                                JTINIT12
      DATA CPCO2/6.214,5.776E-3,-1.094E-6/                              JTINIT13
      DATA CPH2O/7.256,1.277E-3,8.735E-8/                               JTINIT14
      DATA TAB1/1HX,2HXD,2HRD,3HYCB,4HXPRN/                             JTINIT15
      DATA TAD1/1HY,2HUD,3HTID,3HTHD/                                   JTINIT16
    1 MCHANG=.TRUE.                                                     JTINIT17
      NMSH=71                                                           JTINIT18
      CK=1.06064475                                                     JTINIT19
      DY1=.001                                                          JTINIT20
      CXPC=.02                                                          JTINIT21
      CXTP=.02                                                          JTINIT22
      NRED=10                                                           JTINIT23
      DATA BITS1/O377777777777/                                         JTINIT24
      BITS=BITS1                                                        JTINIT25
    2 CALL SETM(1,BITS,MJET,10)                                         JTINIT26
      PF=14.69594                                                       JTINIT27
      TF=518.688                                                        JTINIT28
      TIJET=0.                                                          JTINIT29
      TIE=0.                                                            JTINIT30
      TWO=.FALSE.                                                       JTINIT31
      CALL SETM(1,BITS,MJETO,4)                                         JTINIT32
      TIJETO=0.                                                         JTINIT33
      NJO=50                                                            JTINIT34
      MERGE=.FALSE.                                                     JTINIT35
      AXI=.TRUE.                                                        JTINIT36
      NJ=30                                                             JTINIT37
      NM=80                                                             JTINIT38
      GAM=1.4                                                           JTINIT39
      RG=53.34                                                          JTINIT40
      PR=.72                                                            JTINIT41
      PRT=1.                                                            JTINIT42
      SC=201.6                                                          JTINIT43
      TREF=0.                                                           JTINIT44
      MUREF=0.                                                          JTINIT45
C*                                                                      JTINIT46
    3 OJET=.TRUE.                                                       JTINIT47
      TURBJ=.TRUE.                                                      JTINIT48
C*                                                                      JTINIT49
    4 ERR=.FALSE.                                                       JTINIT50
```

```
      GC=32.174                                                         JTINIT51
      GCJ=25036.442                                                     JTINIT52
      FOOT=12.                                                          JTINIT53
      CORE=.FALSE.                                                      JTINIT54
      SP=1.                                                             JTINIT55
      SV=1.                                                             JTINIT56
      SLEN=1.                                                           JTINIT57
C*                                                                      JTINIT58
    5 CALL SETM(2,BITS,X,100,Y,200)                                     JTINIT59
      CALL SETM(1,0,XPRN,100)                                           JTINIT60
      DPRIN=.FALSE.                                                     JTINIT61
C*                                                                      JTINIT62
      CALL SETM(1,0.,PM,10)                                             JTINIT63
      IMP(1)=0                                                          JTINIT64
      PLOT=0                                                            JTINIT65
      ENTRY1=.TRUE.                                                     JTINIT66
      MIX=.FALSE.                                                       JTINIT67
      CF=.002                                                           JTINIT68
      MAXIT=25                                                          JTINIT69
      SUPR=.TRUE.                                                       JTINIT70
      UPSTRM=.TRUE.                                                     JTINIT71
      CVG=.FALSE.                                                       JTINIT72
      TOL=1.E-6                                                         JTINIT73
      CALL SETM(1,0.,YCB,100)                                           JTINIT74
      CALL SETM(4,BITS,RD,100,XD,100,YR,100,YCD,100)                    JTINIT75
      CALL SETM(1,0,INDC,100)                                           JTINIT76
      NB=5                                                              JTINIT77
      ND    = 4                                                         JTINIT78
      CALL MOVF(2,TAB1,TAB,5,1,TAD1,TAD,4,1)                            JTINIT79
      NC=3                                                              JTINIT80
      CALL MOVF(1,CTABN,CNAME,6,1)                                      JTINIT81
      DIFF=.FALSE.                                                      JTINIT82
      CALL SETM(2,BITS,ALJ,18,ALX,1200)                                 JTINIT83
      CALL SETM(1,.7,SCM,6)                                             JTINIT84
      ALE(1)=.99934                                                     JTINIT85
      ALE(2)=.00033                                                     JTINIT86
      ALE(3)=.00033                                                     JTINIT87
      ALJ(1)=.92                                                        JTINIT88
      ALJ(2)=.04                                                        JTINIT89
      ALJ(3)=.04                                                        JTINIT90
      ALJJ(1)=.96                                                       JTINIT91
      ALJJ(2)=.02                                                       JTINIT92
      ALJJ(3)=.02                                                       JTINIT93
      CALL SETM(1,0.,HCPRE,24)                                          JTINIT94
      CALL SETM(1,600.,TCPRF,3)                                         JTINIT95
      CALL MOVF(4,HH,HCPRF,3,1,CPA,CPC(1,1),3,1,CPC02,CPC(1,2),3,1,     JTINIT96
     * CPH2O,CPC(1,3),3,1)                                              JTINIT97
      RESTRT=BITS                                                       JTINIT98
      NRFS=1                                                            JTINIT99
      MIXPRE=.FALSE.                                                    JTINIT00
```

```
      SCLD   = .FALSE.                                           JTINIT01
      ALXLIM= .0001                                              JTINIT02
      CJRMOD= 1.                                                 JTINIT03
      RETURN                                                     JTINIT04
      END                                                        JTINIT05
```

```
CJTOUT1          JET/OUTPUT AT STATION 1                                        JTOUT101
      SUBROUTINE JTOUT1                                                         JTOUT102
C***********************************************************************JTOUT103
C********************** SPECIAL VERSION FOR 12 SPECIES ******************JTOUT104
C***********************************************************************JTOUT105
      LOGICAL  EOF  ,  ERR                                                      JTOUT106
      LOGICAL  AXI , XPRN , CMPRS , QJET , TURBJ , CORE                         JTOUT107
      REAL  MJET , ME , MUREF                                                   JTOUT108
      COMMON /JETTWO/                                                           JTOUT109
     * TWO    , DIAO    , MJETO  , TJETO  , VJETO  ,                            JTOUT110
     * PTJETO , TIJETO , NJO                                                    JTOUT111
      REAL  MJETO,MACHO                                                         JTOUT112
      COMMON /DIFEQI/                                                           JTOUT113
     * NC , CNAME(12) , ALJ(12) , ALJO(12) ,ALE(12) , SCM(12) ,                 JTOUT114
     * TCPRF(12) , HCPRF(12) , CPC(3,12)                                        JTOUT115
      COMMON /DICTRL/ DIFF , CND(10)                                            JTOUT116
      LOGICAL          DIFF                                                     JTOUT117
      COMMON /MOLES / ALX(100,12)                                               JTOUT118
      COMMON /BCO/  UO, EO, THO                                                 JTOUT119
      COMMON /CTRL2/                                                            JTOUT120
     * EDGEI  , SFI    , MERGE  , XMERGE , YMERGE ,                             JTOUT121
     * SLOPEI , SLOPEO , CEPTI  , CEPTO                                         JTOUT122
      COMMON /MERGET/  MER, MERSTP , XMRG                                       JTOUT123
      LOGICAL  TWO, MERGE , MER , MERSTP                                        JTOUT124
C*                                                                              JTOUT125
C****** INPUT COMMON                                                            JTOUT126
C*                                                                              JTOUT127
      COMMON /INPJET/                                                           JTOUT128
     * DIAJ       , MJET       , TJET       , PTJET      , VJET       ,         JTOUT129
     * TIJET      ,                                                             JTOUT130
     * PE         , VE         , ME         , TIE        , TE         ,         JTOUT131
     * AXI        , NJ         , NM         ,                                   JTOUT132
     * X(100)     , XPRN(100)  ,                                                JTOUT133
     * GAM        , RG         , PR         , PRT        ,                      JTOUT134
     * SC         , TREF       , MUREF                                          JTOUT135
C*                                                                              JTOUT136
C****** CONTROL COMMON                                                          JTOUT137
C*                                                                              JTOUT138
      COMMON /CTRL/                                                             JTOUT139
     * NXTA       , CMPRS      , QJET       , TURBJ      , COEF(10)   ,         JTOUT140
     * NPU  ,  NPD  , DXC  , XU  , XDD  ,                                       JTOUT141
     * DSTOR(800)                                                               JTOUT142
C*                                                                              JTOUT143
C****** PROFILE COMMON                                                          JTOUT144
C*                                                                              JTOUT145
      COMMON /PROF/ PSI(200),Y(200),UD(200),THD(200),ED(200)                    JTOUT146
C*                                                                              JTOUT147
C****** CONSTANT AND ERROR COMMON                                               JTOUT148
C*                                                                              JTOUT149
      COMMON /CNERR/ BITS , ERR , GC , GCJ , FOOT                               JTOUT150
```

```
C*                                                                  JTOUT151
C****** BOUNDARY CONDITION COMMON                                   JTOUT152
C*                                                                  JTOUT153
      COMMON /BC/ UEDGE  , EEDGE  , THEDGE                          JTOUT154
C*                                                                  JTOUT155
C****** POTENTIAL CORE COMMON                                       JTOUT156
C*                                                                  JTOUT157
      COMMON /CORED/ XDCORE  , CORE  , CORSTP                       JTOUT158
C*                                                                  JTOUT159
C****** SCALER (UNITS CONVERSION) COMMON                            JTOUT160
C*                                                                  JTOUT161
      COMMON /SCALER/  SP  , SV  , SLEN                             JTOUT162
C*                                                                  JTOUT163
C****** JET PROPERTIES COMMON                                       JTOUT164
C*                                                                  JTOUT165
      COMMON /JET1/ FLOWJ,TT0,NX,PJET                               JTOUT166
      COMMON /TAG/                                                  JTOUT167
     * NAME(10),TITLE(10),IDENT(10),ADDRES(10),IDENT1(10)           JTOUT168
C*                                                                  JTOUT169
      COMMON /MIXER/ MIX,RD(100),XD(100),CF,YR(100)                 JTOUT170
      LOGICAL  MIX                                                  JTOUT171
      COMMON /FLOBAL/ MAXIT,SUPB,NIT,PSID,YDD,YDC,                  JTOUT172
     *                P1,P2,UCL,TOL,UPSTRM,CVG                      JTOUT173
      LOGICAL  SUPB,CVG,UPSTRM                                      JTOUT174
      COMMON /ACONVG/ YCD(100),PD(100),INDC(100),  CHOKE, CHOKED    JTOUT175
      LOGICAL  CHOKE, CHOKED                                        JTOUT176
      COMMON /DFIT/  CLSP(100)                                      JTOUT177
      COMMON /STA2/ MACH2,TS2,SS2,V2,RHO2,OPDX2                     JTOUT178
      REAL  MACH2                                                   JTOUT179
      COMMON /BCMIX2/ GRADU,TW,MUW,RHOW,PTP,TTP                     JTOUT180
      REAL  MUW                                                     JTOUT181
C*                                                                  JTOUT182
      DIMENSION HEAD1(2,2), HEAD2(2,2), HEAD3(2,2),                 JTOUT183
     * FORM1(2),FORM2(2),FORM3(2), TI(200)                          JTOUT184
      DIMENSION HEAD4(2,3),FORM4(2)                                 JTOUT185
      DIMENSION HEAD5(2,2),FORM5(2)                                 JTOUT186
      EQUIVALENCE (TI(1), DSTOR(1))                                 JTOUT187
      EQUIVALENCE (KAXI,AXI), (KQ,QJET), (KMPRS,CMPRS)              JTOUT188
      EQUIVALENCE  (KTOW,TWO)                                       JTOUT189
      EQUIVALENCE  (IMIX, MIX)                                      JTOUT190
      DATA                                                          JTOUT191
     * HEAD1(1,1)/24HPLANE       AXISYMMETRIC/,                     JTOUT192
     * HEAD2(1,1)/24HISOTHERMAL  N-ISOTHERMAL/,                     JTOUT193
     * HEAD3(1,1)/24HINCOMPRESS. COMPRESSIBLE/                      JTOUT194
      DATA                                                          JTOUT195
     * HEAD4(1,1)/36HSINGLE      CO-PLANAR   CO-ANNULAR  /          JTOUT196
      DATA                                                          JTOUT197
     *HEAD5(1,1)/24HFREE      JETCONFINED JFT/                      JTOUT198
      DATA BLANK/6H      /                                          JTOUT199
C*                                                                  JTOUT100
```

```
C*   SET VARIABLE HEADINGS
C*
     1 NAXI=KAXI+1
       NQ=KQ+1
       NCMP=KMPRS+1
       NJETT=NAXI+KTOW
       IF(NJETT.EQ. 2  .AND.  AXI)  NJETT=1
       KMIX= IMIX+1
C*
C*   CONSTRUCT HOLLERITH HEADINGS
C*
       DO 2 L=1,2
       FORM1(L)=HEAD1(L,NAXI)
       FORM2(L)=HEAD2(L,NQ)
       FORM4(L)=HEAD4(L,NJETT)
       FORM5(L)=HEAD5(L,KMIX)
     2 FORM3(L)=HEAD3(L,NCMP)
C*
C*
       XTREF=TREF
       XMUREF=MUREF
       XSC=SC
       IF(TREF .NE. 0.) GO TO 3
       XTREF=BITS
       XMUREF=BITS
       XSC=BITS
C*
C*   WRITE FIRST SECTION OF OUTPUT
C*
     3 WRITE(6,100)FORM5,FORM1,FORM2,FORM3,FORM4,
      * NAME,ADDRES,TITLE,IDENT,IDENT1,
      * TE,DIAJ,GAM,PE,MJET,RG,
      * VE,TJET,PR,ME,PTJET,PRT,TIE,VJET,XSC,TIJET,
      * XTREF,FLOWJ,XMUREF
       IF(TWO) WRITE (6,105) DIAO,MJETO,TJETO,PTJETO,VJETO,TIJETO
       IF(.NOT.  TWO)  WRITE (6,110)
C*
C*   CONVERT ED TO TURBULENCE INTENSITY
C*
     4 CONV=SQRT(2.*GCJ*EJET/3.)/VJET
       DO 5 L=1,NPU
     5 TI(L)=CONV*SQRT(ED(L))
C*   CHECK FOR DIFFUSION CASE
C*
       IF(DIFF) GO TO 25
C*
C*   WRITE 2=ND SECTION OF OUTPUT
C*
    10 WRITE (6,150)
C*
```

```
C*
   20 WRITE (6,200) (L,Y(L),PSI(L),UD(L),THD(L),TI(L),L=1,NPU)
      GO TO 30
C*
   25 WRITE (6,151) (CNAME(L),L=1,6)
      WRITE (6,201) (L,Y(L),PSI(L),UD(L),THD(L),TI(L),
     * (ALX(L,IL),LL=1,6),L=1,NPU),
      IF( NC.LE.6 ) GO TO 30
C*
C*   SPCEIAL OUTPUT FOR NC.GT. 6
C*
      NC1   = NC+1
      DO 29 K=NC1,12
   29 CNAME(K)= BLANK
      NUM   = (12-NC)*100
      CALL SETH(1,BITS,ALX(1,NC1),NUM)
      WRITE (6,152) (CNAME(L),L=7,12)
      WRITE (6,202) (L,Y(L),PSI(L),(ALX(L,IL),LL=7,12),L=1,NPU)
C*
C*  IF CONFINED MIXING,PRINT PSI OF DUCT SURFACE
C*
   30 IF(MIX) WRITE (6,250) YR(1),PSID
C*
C*********************           FORMAT STATEMENTS          ****************
C*
  100 FORMAT(1H1,23X,3H*   ,2A6,11H PROGRAM  *///,
     * 5H *    ,2A6,11H     *  *    ,2A6,10H    *  *   ,2A6,
     * 10H      * JET///,
     * 16X,6H**     ,2A6,2X,20HMIXING  REGION     **///,
     * 10A6/,10A6/,10A6/,10A6/,10A6///,
     * 17X,37H*  INPUT  AND  INITIAL  CONDITIONS  *//,
     * 2X,19HEXTERNAL CONDITIONS,3X,24HJET DISCHARGE PARAMETERS,
     * 5X,14HGAS PROPERTIES//,2X,4HTE =,F14.3,6X,
     * 6HDIAJ =,F13.5,5X,6HGAM  =,F13.5/,2X,
     * 4HPE =,1PF14.4,6X,6HMJET =,0PF13.4,5X,
     * 6HRG   =,F13.5/,2X,4HVE =,F14.3,6X,6HTJET =,F13.3,
     * 5X,6HPR   =,F13.5/,2X,4HME =,F14.4,6X,6HPTJET=,
     * 1PE13.4,5X,6HPRT  =,0PF13.5/,2X,4HTIE=,1PE14.4,
     * 6X,6HVJET =,0PF13.3,5X,6HSC   =,F13.3/,26X,
     *6HIIJET=,1PE13.4,5X,6HTREF =,0PF13.3/,26X,6HFLOWJ=,1PE13.4,5X,
     * 6HMURFF=,F13.4/)
  110 FORMAT(///)
  105 FORMAT(26X,7HDIAO  =,F12.5/26X,7HMJETO =,F12.4/
     * 26X,7HTJETO =,F12.3/26X,7HPTJETO=,1PE12.4/
     * 26X,7HVJETO =,0PF12.3/26X,7HTIJETO=,1PE12.4//)
C*
  150 FORMAT(23X,26H*    INITIAL    PROFILES    *//
     * 1X,1HN,7X,1HY,11X,3HPSI,13X,1HU,11X,5HTHETA,10X,2HTI//)
C*
  151 FORMAT(53X,26H*    INITIAL    PROFILES    *//
```

```
     *1X,1HN,7X,1HY,11X,3HPSI,10X,1HU,8X,5HTHETA,9X,2HTI,
     * 7X,A6,5X,A6,5X,A6,5X,A6,5X,A6,5X,A6//)
  152 FORMAT(1H1,32X,26H*    INITIAL   PROFILES    *//
     * 1X,1HN,7X,1HY,11X,3HPSI,7X,A6,5(5X,A6)//)
  200 FORMAT(I3,1PE12.4,E13.4,0PF14.8,F14.8,1PE16.7)
  201 FORMAT(I3,1PE12.4,E13.4,0PF11.8,F11.8,1PE15.7,6E11.4)
  202 FORMAT(I3,1PE12.4,E13.4,6E11.4)
  250 FORMAT(//4HWALL,1PE11.4,E13.4)
C*
 1000 RETURN
      END
```

```
CJTOUTS          SUMMARY PRINT OF JET PROPERTIES
      SUBROUTINE JTOUTS
C*******************************************************************************
C*********************** SPECIAL VERSION FOR 12 SPECIES ************************
C*******************************************************************************
      LOGICAL  EOF  ,  ERR
      LOGICAL TAPIN,TAPOT
      LOGICAL  AXI , XPRN , CMPRS , QJET , TURBJ , CORE
      LOGICAL      ENDJOB,ENDJW
      INTEGER XPRN1(100)
      EQUIVALENCE (XPRN,XPRN1)
      INTEGER PLOT
      REAL  MJET , ME , MUREF
      COMMON /ADAM02/ ENDJOB  , DUM02(3)
      COMMON /DIFEQI/
     * NC , CNAME(12) , ALJ(12) , ALJO(12) ,ALE(12) , SCM(12) ,
     * TCPRF(12) , HCPRF(12) , CPC(3,12)
      COMMON /DICTRL/ DIFF , CND(10)
      LOGICAL          DIFF
      COMMON /MOLES / ALX(100,12)
      COMMON /BCMOL / ALEDGE(12),ALO(12)
      COMMON /JETTWO/
     * TWO    , DIAO   , MJETO  , TJETO   , VJETO   ,
     * PTJETO , TIJETO , NJO
      REAL  MJETO,MACHO
      COMMON /BCO/  UO, EO, THO
      COMMON /CTRL2/
     * EDGFI  , SFI    , MERGE  , XMERGE , YMERGE ,
     * SLOPEI , SLOPEO , CEPTI  , CEPTO
      COMMON /MERGET/  MER, MERSTP , XMRG
      LOGICAL  TWO, MERGE , MER , MERSTP
      COMMON /IOFILE/ TAPIN,TAPOT
      COMMON /PARAM/
     * U(200),T(200),TOT(200),XMACH(200),PTOT(200),TTD(200),
     * PTD(200),UDC(200),DUMP9(209)
C*
C****** INPUT COMMON
C*
      COMMON /INPJET/
     * DIAJ        , MJET        , TJET        , PTJET       , VJET        ,
     * TIJET       ,
     * PE          , VE          , ME          , TIF         , TE          ,
     * AXI         , NJ          , NM          ,
     * X(100)      , XPRN(100)   ,
     * GAM         , RG          , PR          , PRT         ,
     * SC          , TREF        , MUREF
C*
C****** CONTROL COMMON
C*
      COMMON /CTRL/
```

```
     * NXTA      , CMPRS     , QJET      , TURBJ      , COEF(10)   , JTOUTS51
     * NPU  ,  NPD  , DXC  , XU  , XDD  ,                               JTOUTS52
     * DSTOR(800)                                                       JTOUTS53
C*                                                                      JTOUTS54
C****** PROFILE COMMON                                                  JTOUTS55
C*                                                                      JTOUTS56
      COMMON /PROF/ PSI(200),Y(200),UD(200),THD(200),ED(200)            JTOUTS57
C*                                                                      JTOUTS58
C****** CONSTANT AND ERROR COMMON                                       JTOUTS59
C*                                                                      JTOUTS60
      COMMON /CNERR/ BITS , ERR , GC , GCJ , FOOT                       JTOUTS61
C*                                                                      JTOUTS62
C****** BOUNDARY CONDITION COMMON                                       JTOUTS63
C*                                                                      JTOUTS64
      COMMON /BC/ UEDGE  , EEDGE  , THEDGE                              JTOUTS65
C*                                                                      JTOUTS66
C****** POTENTIAL CORE COMMON                                           JTOUTS67
C*                                                                      JTOUTS68
      COMMON /CORED/ XCORE  , CORE  , CORSTP                            JTOUTS69
C*                                                                      JTOUTS70
C****** SCALER (UNITS CONVERSION) COMMON                                JTOUTS71
C*                                                                      JTOUTS72
      COMMON /SCALER/  SP  , SV  , SLEN                                 JTOUTS73
C*                                                                      JTOUTS74
C****** JET PROPERTIES COMMON                                           JTOUTS75
C*                                                                      JTOUTS76
      COMMON /JET/                                                      JTOUTS77
     * B(100)  , UC(100)  , TC(100)  , TIC(100) ,                       JTOUTS78
     * PTC(100) , WJ(100)  , YJ(100)  ,                                 JTOUTS79
     * YSONIC(100)                                                      JTOUTS80
      COMMON /JET1/ FLOWJ,TTO,NX,EJET                                   JTOUTS81
      COMMON /JET2/  TTC(100)                                           JTOUTS82
      COMMON /TAG/                                                      JTOUTS83
     * NAME(10),TITLE(10),IDENT(10),ADDRES(10),IDENT1(10)               JTOUTS84
      COMMON /MISC/  PM(10),  PLOT                                      JTOUTS85
C*                                                                      JTOUTS86
      COMMON /MIXER/ MIX,RD(100),XD(100),CF,YR(100)                     JTOUTS87
      LOGICAL  MIX                                                      JTOUTS88
      COMMON /FLOBAL/ MAXIT,SUPB,NIT,PSID,YDD,YDC,                      JTOUTS89
     *              P1,P2,UCL,TOL,UPSTRM,CVG                            JTOUTS90
      LOGICAL  SUPB,CVG,UPSTRM                                          JTOUTS91
      COMMON /ACONVG/ YCD(100),PD(100),INDC(100),  CHOKE, CHOKED        JTOUTS92
      LOGICAL  CHOKE, CHOKED                                            JTOUTS93
      COMMON /DFTT/  CLSP(100)                                          JTOUTS94
      COMMON /STA2/ MACH2,TS2,SS2,V2,RHO2,DPDX2                         JTOUTS95
      REAL  MACH2                                                       JTOUTS96
      COMMON /BCMIX2/ GRADU,TW,MUW,RHOW,PTE,TTE                         JTOUTS97
      REAL  MUW                                                         JTOUTS98
      COMMON /THERM/ GMC(200),CP(200)                                   JTOUTS99
      COMMON /MIXPRP/ MA2(100),VE2(100),TE2(100),TWC(100)               JTOUTS00
```

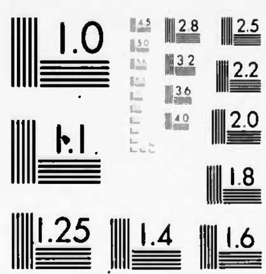
```
      REAL MA2
      COMMON /THRST/  WV(100)
      COMMON /CBODY/  YCB(100),CLSPCB(100),YCB1 , UCL1
      COMMON /OUTMIX/ NXORIG
      COMMON /JET3/ STADD, NV, STATF
      LOGICAL  STADD,STATF
C*
      DIMENSION TI(200)
      DIMENSION LF1(3)
      DIMENSION  HEAD1(2,2), FORM1(2)
      EQUIVALENCE (TI(1),DSTOR(1))
      EQUIVALENCE  (IMIX, MIX)
      DATA LF1(1)/18HSUBSONSUPSONCHOKED/
      DATA HEAD1(1,1)/24HFREE JET     CONFINED JET/
      DATA ENDJW/F/
C*
      GO TO 1
      ENTRY JTPWR
      NXTA=NX-1
      PLOT=0
      ENDJW=.TRUE.
    1 IF(PLOT.GT.  0    .OR.    TAPOT) CALL JTFILE(2,DUM)
      KMIX= IMIX+1
      DO 6 L=1,2
    6 FORM1(L)=HEAD1(L,KMIX)
      DO 7 L=1,NXTA
      IF(INDC(L).EQ.0) INDC(L)=LF1(1)
      IF(INDC(L).EQ.1) INDC(L)=LF1(2)
      IF(INDC(L).EQ.2) INDC(L)=LF1(3)
    7 CONTINUE
      TERMD=.5*DIAJ
      CALL FMPYC(1,TERMD,YCB,YCB,NXORIG)
      WRITE (6,100)
      NSTART=1
      NLINES=NXTA
      IF(NLINES.GT.50) NLINES=50
      NL=NLINES
    2 WRITE (6,110) FORM1,IDENT,IDENT1
    3 WRITE (6,120)
      NEND=NL
      ASSIGN 10 TO LGO
    5 WRITE (6,200) (L,X(L),B(L),YJ(L),UC(L),TC(L),
     * TIC(L),PTC(L),TTC(L),YSONIC(L),WJ(L),L=NSTART,NEND)
      GO TO LGO  , (10,20)
C*
C*** CHECK FOR ADDITIONAL LINES
C*
   10 IF(NXTA.LE.50) GO TO 20
      NSTART=NEND+1
      NEND=NXTA
```

AD-A045 627 GENERAL ELECTRIC CO CINCINNATI OHIO AIRCRAFT ENGINE GROUP F/G 21/5
DEVELOPMENT OF EMISSIONS MEASUREMENT TECHNIQUES FOR AFTERBURNIN--ETC(U)
OCT 75 W C COLLEY, D R FERGUSON, M A SMITH F33615-73-C-2047
UNCLASSIFIED R75AEG459 AFAPL-TR-75-52-SUPPL-2 NL

4 OF 5
ADA045 627

4 OF

ADA045627



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

```
      WRITE (6,100)                                                     JTOUTS51
      WRITE (6,120)                                                     JTOUTS52
      ASSIGN 20 TO LGO                                                  JTOUTS53
C*                                                                      JTOUTS54
      GO TO 5                                                           JTOUTS55
C*                                                                      JTOUTS56
C*  IF CONFINED JET CASE, PRINT CONFINED JET OUTPUT                     JTOUTS57
C*                                                                      JTOUTS58
   20 IF(.NOT. MIX) GO TO 21                                            JTOUTS59
      WRITE (6,100)                                                     JTOUTS60
      NSTART=1                                                          JTOUTS61
      NLINES=NV                                                         JTOUTS62
      IF(NLINES.GT.50) NLINES=50                                        JTOUTS63
      NL=NLINES                                                         JTOUTS64
   22 WRITE (6,110) FORM1,IDENT,IDENT1                                  JTOUTS65
   23 WRITE (6,140)                                                     JTOUTS66
      NEND=NL                                                           JTOUTS67
      ASSIGN 30 TO LGO                                                  JTOUTS68
   25 WRITE(6,2000) (L,XD(L),RD(L),YCB(L),YR(L),YCD(L),                 JTOUTS69
     * PD(L),INDC(L),WV(L),MA2(L),VE2(L),TE2(L),L=NSTART,NEND)          JTOUTS70
      GO TO  LGO , (30,21)                                              JTOUTS71
C*                                                                      JTOUTS72
   30 IF(NV.LE.50) GO TO 21                                             JTOUTS73
      NSTART=NEND+1                                                     JTOUTS74
      NEND=NV                                                           JTOUTS75
      WRITE (6,100)                                                     JTOUTS76
      WRITE (6,140)                                                     JTOUTS77
      ASSIGN  21 TO LGO                                                 JTOUTS78
      GO TO 25                                                          JTOUTS79
C*                                                                      JTOUTS80
   21 IF(.NOT. ENDJW)  RETURN                                           JTOUTS81
      ENDJOB=.TRUE.                                                     JTOUTS82
      CALL ERROR                                                        JTOUTS83
C*                                                                      JTOUTS84
C****************************   FORMAT STATEMENTS    ****************** JTOUTS85
C*                                                                      JTOUTS86
  100 FORMAT (1H1)                                                      JTOUTS87
  110 FORMAT(39X,49H*   J E T    A N A L Y S I S    P R O G R A M   *//,JTOUTS88
     * 48X,2A6,6X,6HMIXING//                                            JTOUTS89
     * 30X,10A6/,30X,10A6//)                                            JTOUTS90
  120 FORMAT(38X,51H*   SUMMARY  -  STATION DATA  -  JET PROPERTIES   * JTOUTS91
     *//,4X,1HN,10X,1HX,13X,1HB,11X,2HYJ,12X,2HUC,                      JTOUTS92
     * 12X,2HTC,11X,3HTIC,7X,3HPTC,8X,3HTTC,5X,6HYSONIC,6X,2HWJ//)      JTOUTS93
  200 FORMAT(3X,I3,F14.5,F13.5,F14.5,F15.7,F14.7,1PE15.6,               JTOUTS94
     * 0PF9.6,3F10.6)                                                   JTOUTS95
  140 FORMAT(4X,1HN,6X,2HXD,8X,2HRD,7X,3HYCB,8X,2HYD,9X,3HYCD,          JTOUTS96
     * 9X,2HPD,5X,4HFLOW,2X,6HTHRUST,6X,3HMA2,5X,3HVE2,                 JTOUTS97
     * 6X,3HTE2//)                                                      JTOUTS98
 2000 FORMAT(3X,I3,3F10.4,2F12.7,F9.4,1X,A6,F11.3,F7.4,F9.3,F8.2)       JTOUTS99
C*                                                                      JTOUTS00
```

```
C*
C******************************   JTOUTP ENTRY   ***********************
C*
C*   PROFILE PRINTOUT
C*
      ENTRY JTOUTP
      PK=PE
      IF(MIX) PK=P2
C*
C*
   50 NSTA=NX
   51 XSTA=X(NX)
C*
C*  CONVERT TURBULENCE ENERGIES TO INTENSITIES
C*
   52 CON1=SQRT(2.*GCJ*EJET/3.)/VJET
      DO 53 L=1,NPD
   53 TI(L)=CON1*SQRT(ED(L))
C*
C*  COMPUTE ADDITIONAL DIMENSIONLESS AND DIMENSIONAL PROFILES
C*
      UC1=1./UC(NX)
      TMPT=1./(TTO-TE)
      PTPT=1./(PTJET-PE)
      CALL FMPYC(1,TJET,THD,T,NPD)
      CALL GAMCP(T,GMC,CP,RG,1,NPD)
      DO 600 L=1,NPD
      U(L)=VJET*UD(L)
      IF(.NOT. CMPRS)  GO TO 600
      GAM=GMC(L)
      GM2=2./(GAM-1.)
      GMM=GAM/(GAM-1.)
      CPJ=CP(L)
      GCJCP=1./(GCJ*CPJ)
      TOT(L)=T(L)+.5*U(L)*U(L)*GCJCP+EJET*ED(L)/CPJ
      XMACH(L)=U(L)/SQRT(GAM*RG*GC*T(L))
      PTOT(L)=PK*(1.+XMACH(L)**2/GM2 )**GMM
      TTD(L)=(TOT(L)-TE)*TMPT
      PTD(L)=(PTOT(L)-PE)*PTPT
  600 CONTINUE
      IF(.NOT. CMPRS)  CALL SETM(5,BITS,TOT,NPD,XMACH,NPD,
     * PTOT,NPD,TTD,NPD,PTD,NPD)
C*
      IF(PLOT.GT.0  .OR.  (XPRN(NX)  .AND.  TAPOT)) CALL JTFILE(1,XSTA)
      IF(XPRN1(NX)  .LT. 0)  GO TO 111
   55 WRITE (6,300)
      NSTART=1
      NLINES=NPD
      IF(NLINES.GT.50) NLINES=50
      NL=NLINES
```

```
   56 WRITE (6,310) NSTA,XSTA,PK                                          JTOUTS51
      NEND=NL                                                             JTOUTS52
      ASSIGN 60 TO LGO1                                                   JTOUTS53
   57 WRITE (6,320) (L,Y(L),PSI(L),UD(L),THD(L),TI(L),TTD(L),PTD(L),      JTOUTS54
     * XMACH(L),U(L),T(L),TOT(L),PTOT(L),L=NSTART,NEND)                   JTOUTS55
      GO TO LGO1  , (60,90)                                               JTOUTS56
C*                                                                        JTOUTS57
C*  CHECK FOR ADDITIONAL PRINT LINES                                      JTOUTS58
C*                                                                        JTOUTS59
   60 IF(NPD.LE. 50) GO TO 90                                             JTOUTS60
      WRITE (6,100)                                                       JTOUTS61
      WRITE (6,310) NSTA,XSTA,PK                                          JTOUTS62
      NRMN=NPD-NL                                                         JTOUTS63
      IF(NRMN.GT.50) GO TO 65                                             JTOUTS64
      NSTART=NEND+1                                                       JTOUTS65
      NEXT=MIN0(50,NRMN)                                                  JTOUTS66
      NL=NL+NEXT                                                          JTOUTS67
      ASSIGN 90 TO LGO1                                                   JTOUTS68
      NEND=NL                                                             JTOUTS69
      GO TO 57                                                            JTOUTS70
   65 NSTART=NEND+1                                                       JTOUTS71
      NEND=NL+50                                                          JTOUTS72
      NL=NL+50                                                            JTOUTS73
      GO TO 57                                                            JTOUTS74
C*                                                                        JTOUTS75
C*                                                                        JTOUTS76
C*  PRINT CONCENTRATION PROFILES                                          JTOUTS77
C*                                                                        JTOUTS78
   90 IF(.NOT. DIFF) GO TO 1111                                           JTOUTS79
      WRITE (6,300)                                                       JTOUTS80
      NSTART=1                                                            JTOUTS81
      NLINES=NPD                                                          JTOUTS82
      IF(NLINES.GT.50) NLINES=50                                          JTOUTS83
      NL=NLINES                                                           JTOUTS84
 1112 WRITE (6,311) NSTA,XSTA,PK,CNAME                                    JTOUTS85
      NEND=NL                                                             JTOUTS86
      ASSIGN 1120 TO LGO1                                                 JTOUTS87
 1113 WRITE (6,321) (L,Y(L),(ALX(L,LL),LL=1,12),L=NSTART,NEND)            JTOUTS88
      GO TO  LGO1  , (1120,1111)                                          JTOUTS89
C*                                                                        JTOUTS90
 1120 IF(NPD.LE.50) GO TO 1111                                            JTOUTS91
      WRITE (6,300)                                                       JTOUTS92
      WRITE (6,311) NSTA,XSTA,PK,CNAME                                    JTOUTS93
      NRMN=NPD-NL                                                         JTOUTS94
      IF(NRMN.GT.50) GO TO 1114                                           JTOUTS95
      NSTART=NEND+1                                                       JTOUTS96
      NEXT=MIN0(50,NRMN)                                                  JTOUTS97
      NL=NL+NEXT                                                          JTOUTS98
      ASSIGN 1111 TO  LGO1                                                JTOUTS99
      NEND=NL                                                             JTOUTS00
```

```
      GO TO  1113                                                       JTOUTS01
 1114 NSTART=NL+50                                                      JTOUTS02
      NEND=NL+50                                                        JTOUTS03
      NL=NL+50                                                          JTOUTS04
      GO TO 1113                                                        JTOUTS05
C****************************  FORMAT STATEMENTS   ******************** JTOUTS06
C*                                                                      JTOUTS07
  300 FORMAT(1H1,50X,28H*  JET  ANALYSIS  PROGRAM  *//)                 JTOUTS08
  310 FORMAT(26X,16HPROFILES-- STA (,I3,6H)   X=,F10.5,                 JTOUTS09
     * 3X,9HPRESSURE=,F10.4//                                           JTOUTS10
     *  22X,1H*,21X,21H**  DIMENSIONLESS  **,22X,                       JTOUTS11
     *  2H**,12X,19H**  DIMENSIONAL   **, 9X,1H*/                       JTOUTS12
     * 2X,1HN,6X,1HY,9X,3HPSI,8X,2HUD,7X,3HTHD,9X,2HTI,8X,3HTTD,8X,     JTOUTS13
     *  3HPTD,6X,4HMACH,8X,1HU,11X,1HT,9X,3HTOT,7X,4HPTOT//)            JTOUTS14
  311 FORMAT(12X,22HMOLE FRACTIONS-- STA (,I3,6H)   X=,                 JTOUTS15
     * F10.5,3X,9HPRESSURE=,F10.4//                                     JTOUTS16
     * 2X,1HN,6X,1HY,5X,12(A6,3X)//)                                    JTOUTS17
  320 FORMAT(1X,I3,F11.5,1PE11.4,0PF9.6,F10.6,1PF12.5,0PF10.7,          JTOUTS18
     *  F11.7,F9.5,3F11.4,F10.4)                                        JTOUTS19
  321 FORMAT(1X,I3,F9.5,12F9.6)                                         JTOUTS20
C*                                                                      JTOUTS21
 1111 RETURN                                                            JTOUTS22
      END                                                               JTOUTS23
```

```
CJTSTP          JET--SOLUTION ROUTINE
      SUBROUTINE JTSTEP
C****************************************************************************
C********************** SPECIAL VERSION FOR 12 SPECIES ***********************
C****************************************************************************
      LOGICAL  ICYCLE
      INTEGER TWOJ, ITWO
      LOGICAL SUPC,SUPSTP
      LOGICAL SUBSON
      LOGICAL TROUBL
      LOGICAL DPRIN
      LOGICAL LAST,CORSTP,ADDP,ENTRY1,IER
      LOGICAL  EOF  ,  ERR
      LOGICAL  AXI , XPRN , CMPRS , QJET , TURBJ , CORE
      REAL KCP,MUL,MUEFF,MACH
      REAL  MJET , ME , MUREF
      COMMON /RSTART/ NREG,RESTRT,NRES,MIXPRE
      LOGICAL  MIXPRE
      COMMON /MOLUP / ALXU(100,12),DALXU(100,12),DTKE(200)
      COMMON /DIFEQI/
     * NC , CNAME(12) , ALJ(12) , ALJD(12) ,ALE(12) , SCM(12) ,
     * TCPRE(12) , HCPRE(12) , CPC(3,12)
      COMMON /DICTRL/ DIFF , CND(10)
      LOGICAL          DIFF
      COMMON /MOLES / ALX(100,12)
      COMMON /BCMOL / ALEDGE(12),ALO(12)
      COMMON /JETTWO/
     * TWO     , DIAO    , MJETO   , TJETO   , VJETO   ,
     * PTJETO , TIJETO , NJO
      REAL  MJETO,MACHO
      COMMON /BCO/  UO, EO, THO
      COMMON /CTRL2/
     * EDGEI   , SEI     , MERGE   , XMERGE , YMERGE ,
     * SLOPEI  , SLOPEO  , CEPTI   , CEPTO
      COMMON /MERGET/  MER, MERSTP , XMRG
      LOGICAL  TWO, MERGE , MER , MERSTP
      COMMON /SETNEW/  LEDGE, LCOREN
      COMMON /INP1  /         ENTRY1
      COMMON /MISC/ PM(10)
      COMMON /PARAM/
     * AL(200)  , BE(200)  , GM(200)  ,
     * EPS(200) , DL(200)  ,
     * VAR(200) , DVAR(200),
     * SM1(200) , NM1      , SM(200)  , NM      ,
     * DX  ,
     * B1  , C11 , D1  ,
     * AN  , BN  , DN
      COMMON /PARAM1/ ETA(200)
C*
C****** INPUT COMMON
```

```
C*
      COMMON /INPJET/
     * DIAJ     , MJET      , TJET      , PTJET     , VJET      ,
     * TIJET    ,
     * PE       , VE        , ME        , TIE       , TE        ,
     * AXI      , NJ        , NMAX      ,
     * XJ(100)  , XPRN(100) ,
     * GAM      , RG        , PR        , PRT       ,
     * SC       , TREF      , MUREF
C*
C****** CONTROL COMMON
C*
      COMMON /CTRL/
     * NXTA     , CMPRS     , QJET      , TURBJ     , COEF(10)  ,
     * NPU  ,  NPD  , DXC  , XU  , XDD  ,
     * DSTOR(800)
C*
C****** PROFILE COMMON
C*
      COMMON /PROF/ PSI(200),Y(200),UD(200),THD(200),ED(200)
C*
C****** CONSTANT AND ERROR COMMON
C*
      COMMON /CNERR/ BITS , ERR , GC , GCJ , FOOT
C*
C****** BOUNDARY CONDITION COMMON
C*
      COMMON /BC/ UEDGE  , EEDGE  , THEDGE
C*
C****** POTENTIAL CORE COMMON
C*
      COMMON /CORED/ XCORE  , CORE  , CORSTP
C*
      COMMON /SUPER/ SUPC,SUPSTP,XSUP
C****** SCALER (UNITS CONVERSION) COMMON
C*
      COMMON /SCALER/  SP  , SV  , SLEN
C*
C****** JET PROPERTIES COMMON
C*
      COMMON /JET/
     * B(100)  , UC(100)  , TC(100)  , TIC(100) ,
     * PTC(100) , WJ(100)  , YJ(100)
      COMMON /JET1/ FLOWJ,TTO,NX,EJET
      COMMON /ERASE/ YD(200),TKE(200),T(200),DUDY(200)
C*
      COMMON /PROPJT/
     * P        , PRL      , PRTT     , RGAS     , SCC      ,
     * TREFF    , VSREF    , MACH     , XLC      ,
     * REFL     , C        , CHI      , RNORM    ,
```

```
     * RHO(200)    , MUL(200)    , KCP(200)    ,                         JTSTP101
     * MUEFF(200)  , XLN(200)    , DK(200)     , RETURB(200)             JTSTP102
       COMMON /XPRIN/ DPRIN                                              JTSTP103
       COMMON /EDGE/  YJETE  ,  SFEDGE                                   JTSTP104
       COMMON /UMFSH/ DUMU1(4),CXPC,CXTP,NRFD                            JTSTP105
C*                                                                       JTSTP106
       COMMON /MIXER/  MIX,RDD(100),XD(100),CF,YR(100)                   JTSTP107
       LOGICAL  MIX                                                      JTSTP108
       COMMON /FLOBAL/ MAXIT,SUPB,NIT,PSID,YDD,YDC,                      JTSTP109
     *                 P1,P2,UCL,TOL,UPSTRM,CVG                          JTSTP110
       LOGICAL  SUPB,CVG,UPSTRM                                          JTSTP111
       COMMON /ACONVG/ YCD(100),PD(100),INDC(100),  CHOKE, CHOKED        JTSTP112
       LOGICAL  CHOKE, CHOKED                                            JTSTP113
       COMMON /DFIT/  CLSP(100)                                          JTSTP114
       COMMON /STA2/ MACH2,TS2,SS2,V2,RHO2,DPDX2                         JTSTP115
       REAL  MACH2                                                       JTSTP116
       COMMON /BCMIX2/ GRADU,TW,MUW,RHOW,PTF,TTE                         JTSTP117
       REAL  MUW                                                         JTSTP118
       COMMON /THERM/ GMC(200),CP(200)                                   JTSTP119
       COMMON /CBODY/  YCB(100),CLSPCB(100),YCB1 , UCL1                  JTSTP120
       COMMON /OUTMIX/ NXORIG                                            JTSTP121
       COMMON /SCALED/ SCLD,ALXLIM                                       JTSTP122
       LOGICAL          SCLD                                             JTSTP123
C*                                                                       JTSTP124
C*                                                                       JTSTP125
       DIMENSION HU(200),EU(200),THU(200)                                JTSTP126
       DIMENSION UK(200)                                                 JTSTP127
       DIMENSION NAM(6)                                                  JTSTP128
       DIMENSION YU(200)                                                 JTSTP129
       EQUIVALENCE (YU(1),DSTOR(401))                                    JTSTP130
       EQUIVALENCE  (UK(1),DSTOR(1))                                     JTSTP131
       EQUIVALENCE                                                       JTSTP132
     * (C1,COEF(1)),(C2,COEF(2)),(C[illegible]),(C4,COEF(4)),            JTSTP133
     * (C5,COEF(5)),(C6,COEF(6)),([illegible]))                          JTSTP134
       EQUIVALENCE (C9,COEF(9))                                          JTSTP135
       EQUIVALENCE (ITWO,IWO)                                            JTSTP136
       DATA NAM/1HY,2HSM,3HXLN,2HUD,?HEC,[illegible]HTHD/                JTSTP137
*      DATA ENTRY1/T/                                                    JTSTP138
C*                                                                       JTSTP139
C*  TEST FOR 1-ST STEP(ENTRY1=T)                                         JTSTP140
C*                                                                       JTSTP141
       ASSIGN 16 TO LGOP                                                 JTSTP142
       IF(ENTRY1 .AND. (RESTRT.EQ.BITS)) CORSTP=.FALSE.                  JTSTP143
     5 IF(ENTRY1) DX=AMIN1(CXPC*B(NRES),.5*DXC)                          JTSTP144
       IF(RESTRT .EQ. BITS) GO TO 5326                                   JTSTP145
       IF(NREG.GT. 1) DX=AMIN1(CXTP*B(NRES),.5*DXC)                      JTSTP146
  5326 IF(DXC.EQ.0.) GO TO 5327                                          JTSTP147
       IF(.NOT. CORSTP) DX=AMIN1(DX, .5*DXC)                             JTSTP148
  5327 CONTINUE                                                          JTSTP149
       IF(.NOT. ENTRY1) GO TO 9                                          JTSTP150
```

```
C*
C*----ENTRY 1--INITIALIZE FOR 1-ST STEP.
C*
      CALL SETM(1,0.,ETA,200)
      ICYCLE=.FALSE.
      TWOJ=ITWO+1
      CALL MOVE(5,PSI,SM,NPU,1,PSI,SM1,NPU,1,UD,UU,NPU,1,THD,THU,NPD,1,
     * ED,EU,NPU,1)
      IF(.NOT. DIFF) GO TO 7000
      DO 7001 L=1,NC
      CALL MOVE(1,ALX(1,L),ALXU(1,L),NPU,1)
 7001 CONTINUE
 7000 NM=NPU
      NM1=NPU
      EC=ED(1)
      THC=THD(1)
      IF(.NOT. TURBJ) CALL SETM(2,EC,ED,200,EU,200)
      IF(.NOT. QJET) CALL SETM(2,THC,THD,200,THU,200)
      EPS1=0.
      IF(AXI) EPS1=1.
C*
    4 LAST=.FALSE.
      ADDP=.FALSE.
      IF(RESTRT.NE. BITS) GO TO 6
      CORSTP=.FALSE.
      SUPSTP=.FALSE.
      MERSTP=.FALSE.
      NREG=1
      SUBSON=.TRUE.
      IF(MJET.GE.1.) SUBSON=.FALSE.
    6 X=XU
      LCORFN=1
      LEDGF=0
      CALL SETM(1,C1,AL,200)
      CALL MOVE(1,Y,YU,NPU,1)
C*
C*  INCREMENT STEP COUNTER, X-STATION ,ETC.
C*  RETURN FOR NEXT STEP IS MADE TO THIS POINT.
C*
    8 NSTP=NSTP+1
    9 X=X+DX
C*
C*  IF COANNULAR PROBLEM,  SAVE UO,THO
C*
      IF(.NOT. TWO)   GO TO 9966
      USVO=UO
      THSVO=THO
 9966 IF (MIX) CALL AITER1(X,DX)
      NHALF=0
      IF(RESTRT.NE.BITS) GO TO 10
```

```
      IF(ENTRY1  .OR.  ICYCLE)  GO TO 20
C*
C*  IF 1-ST STEP PROPERTY CALCULATION IS BYPASSED
C*
   10 CALL FMPYC(1,C6,Y,YD,NPU)
      CALL FMPYC(1,EJET,ED,TKE,NPU)
      CALL FMPYC(1,TJET,THD,T,NPU)
C*
C*  CALL SCALE TO COMPUTE WIDTH OF MIXING ZONE(S)
C*  AND REFERENCE SCALES FOR TURBULENCE
C*
   12 IF( SCLD ) GO TO 13
      CALL SCALE(UU,TWOJ,NREG,X)
      GO TO 15
   13 CALL SCALE(ALXU(1,NC),TWOJ,NREG,X)
C*
C*  COMPUTE PROPERTIES
C*
   15 CALL PROPJ(TWOJ,TURBJ,NREG,X,YD,T,TKE,1,NPU)
      CALL GAMCP(T,GMC,CP,RG,1,NPU)
      GO TO LGOP,  (16,1001)
C*
C*  ADD MESH POINT TO DOWNSTREAM STATION IF ADDP=T.
C*
   16 IF(.NOT.ADDP) GO TO 18
   17 CALL PADD(SM,NPD,NREG)
      ADDP=.FALSE.
   18 NM=NPD
C*
C*  ASSURE THAT UPSTREAM PSI IS FAR ENOUGH OUT
C*
      IF(SM1(NPU).EQ.SM(NPD)) GO TO 20
      SM1(NPU+1)=SM(NPD)
C*
C*  EXTRAPOLATE APPROXIMATE Y
C*  LINEAR EXTRAPOLATION
C*
      NPU=NPU+1
      NM1=NPU
      DYDPSI=(Y(NPU-1)-Y(NPU-2))/(SM1(NPU-1)-SM1(NPU-2))
      Y(NPU)=Y(NPU-1)+DYDPSI*(SM1(NPU)-SM1(NPU-1))
      UU(NPU)=UU(NPU-1)
      THU(NPU)=THU(NPU-1)
      EU(NPU)=EU(NPU-1)
      T(NPU)=TJET*THU(NPU)
      TKE(NPU)=EJET*EU(NPU)
      YD(NPU)=C6*Y(NPU)
      IF(.NOT. DIFF) GO TO 7002
      DO 7003 L=1,NC
 7003 ALXU(NPU,L)=ALXU(NPU-1,L)
```

```
 7002 CALL PROPJ(TWOJ,TURBJ,NREG,X,YD,T,TKF,NPU,NPU)                    JTSTP251
      CALL GAMCP(T,GMC,CP,RG,NPU,NPU)                                   JTSTP252
C*                                                                      JTSTP253
C****  SOLUTION OF MOMENTUM EQUATION                                    JTSTP254
C*                                                                      JTSTP255
   20 CALL MOVE(1,UU,VAR,NM1,1)                                         JTSTP256
C*                                                                      JTSTP257
C*  TEMPORARILY SAVE CURRENT Y VALUES ON ITERATION                      JTSTP258
C*                                                                      JTSTP259
      IF(MIX .AND. (.NOT. ICYCLE))  CALL MOVE(1,Y,YU,NPU,1)             JTSTP260
      CALL SETM(1,1.,EPS,NPU)                                           JTSTP261
      CALL SETM(1,C1,AL,NPU)                                            JTSTP262
      CALL SETM(2,0.,GM,NPU,DL,NPU)                                     JTSTP263
      ENTRY1=.FALSE.                                                    JTSTP264
      IF(RESTRT.NE.BITS)  RESTRT=BITS                                   JTSTP265
C*                                                                      JTSTP266
      IF(.NOT. DPRIN)  GO TO 19                                         JTSTP267
      WRITE (6,8680) NSTP,X,DX                                          JTSTP268
 8680 FORMAT(//6X,5HSTEP=,I4,3X,2HX=,1PE16.8,3X,3HDX=,E16.8//)          JTSTP269
      WRITE (6,8681) YJETE,SFEDGE,EDGEI,SFI                             JTSTP270
 8681 FORMAT(6X,6HYJETE=,1PE16.8,6X,7HSFEDGE=,E16.8/,                   JTSTP271
     * 6X,6HEDGEI=,E16.8,6X,7HSFI    =,E16.8//)                         JTSTP272
      CALL TABPRT(NAM(1),Y,NPU,10,0)                                    JTSTP273
      CALL TABPRT(NAM(2),SM,NPD,10,0)                                   JTSTP274
      CALL TABPRT(NAM(3),XLN,NPD,10,0)                                  JTSTP275
   19 DO 21 L=2,NPU                                                     JTSTP276
      RAD=1.                                                            JTSTP277
      IF(AXI) RAD=Y(L)**2                                               JTSTP278
      IF(MIX)GM(L)=-C7*DPDX2/(RHO(L)*UU(L))                             JTSTP279
   23 BE(L)=MUEFF(L)*RHO(L)*UU(L)*RAD                                   JTSTP280
      IF(.NOT. TURBJ) BE(L)=MUL(L)*RHO(L)*UU(L)*RAD                     JTSTP281
   21 CONTINUE                                                          JTSTP282
      BE(1)=0.                                                          JTSTP283
      IF(.NOT. AXI) BE(1)=MUEFF(1)*RHO(1)*UU(1)                         JTSTP284
C*                                                                      JTSTP285
C*  BOUNDARY CONDITIONS                                                 JTSTP286
C*                                                                      JTSTP287
      B1=0.                                                             JTSTP288
      C11=1.                                                            JTSTP289
      D1=0.                                                             JTSTP290
      AN=1.                                                             JTSTP291
      BN=0.                                                             JTSTP292
   22 DN=UEDGE                                                          JTSTP293
      IF(.NOT.MIX) GO TO 24                                             JTSTP294
C*                                                                      JTSTP295
C*  B.C. FOR CONFINED MIXER                                             JTSTP296
C*                                                                      JTSTP297
      IF(UPSTRM) GO TO 24                                               JTSTP298
      AN=0.                                                             JTSTP299
      DN=GRADU                                                          JTSTP300
```

```
      BN=1.                                                             JTSTP301
C*                                                                      JTSTP302
   24 CALL DFEQ(0,0,0,IER)                                              JTSTP303
      IF(IER) GO TO 999                                                 JTSTP304
C*                                                                      JTSTP305
   25 CALL MOVE(2,VAR,UD,NPD,1,DVAR,DUDY,NPD,1)                         JTSTP306
      IF(UFDGE.EQ.0. .AND. UD(1).GT.1. .AND.(.NOT.SCLD)) UD(1)=1.       JTSTP307
C*                                                                      JTSTP308
C*                                                                      JTSTP309
C*                                                                      JTSTP310
C*  VALUES OF UD WITHIN UD(1)=UCL1,  SET TO UCL1                        JTSTP311
C*                                                                      JTSTP312
      UCL1=UD(1)                                                        JTSTP313
      CLU=UCL1+1.E-6                                                    JTSTP314
      CLL=UCL1-1.E-6                                                    JTSTP315
  100 DO 1111 L=1,NM                                                    JTSTP316
      IF(UD(L).GE.CLL  .AND.  UD(L).LE.CLU)  UD(L)=UCL1                 JTSTP317
 1111 CONTINUE                                                          JTSTP318
      DO 1112 L=1,NPU                                                   JTSTP319
      UK(L)=YOF(SM1(L),SM,UD,1,NPD)                                     JTSTP320
      DUDY(L)=YOF(SM1(L),SM,DVAR,1,NPD)                                 JTSTP321
 1112 CONTINUE                                                          JTSTP322
      IF(DPRIN) CALL TABPRT(NAM(4),UD ,NPD,10,0)                        JTSTP323
C*                                                                      JTSTP324
C*                                                                      JTSTP325
C*                                                                      JTSTP326
C*  TEST FOR TURBULENT PROBLEM                                          JTSTP327
C*                                                                      JTSTP328
   28 IF(.NOT.  TURBJ) GO TO 7010                                       JTSTP329
C*                                                                      JTSTP330
C****  SOLUTION OF TKE EQUATION                                         JTSTP331
C*                                                                      JTSTP332
   30 NM1=NPU                                                           JTSTP333
      CALL MOVE(1,EU,VAR,NPU,1)                                         JTSTP334
      DO 31 L=2,NPU                                                     JTSTP335
      RAD=1.                                                            JTSTP336
      IF(AXI) RAD=Y(L)**2                                               JTSTP337
      RD=RHO(L)*UK(L)                                                   JTSTP338
      BE(L)=MUL(I )*DK(L)*RD*RAD                                        JTSTP339
      DUDYSQ=DUDY(L)*DUDY(L)                                            JTSTP340
      GM(L)=C2*MUL(L)*(MUEFF(L)/MUL(L)-1.)*RD*DUDYSQ*RAD                JTSTP341
      IF(RD.EQ.0.  .OR.  UD(L).LE. .005) GO TO 33                       JTSTP342
      DL(L)=-C5*C*MUL(L)*DK(L)/(RD*XLN(L)*XLN(L))                       JTSTP343
      GO TO 31                                                          JTSTP344
   33 DL(L)=0.                                                          JTSTP345
      GM(L)=0.                                                          JTSTP346
   31 CONTINUE                                                          JTSTP347
      BE(1)=0.                                                          JTSTP348
      IF(.NOT. AXI) BE(1)=MUL(1)*DK(1)*RHO(1)*UD(1)                     JTSTP349
C*                                                                      JTSTP350
```

```
      AN=1.
      BN=0.
   32 DN=EEDGE
      IF(.NOT. MIX)  GO TO 34
      IF(UPSTRM)  GO TO 34
      AN=0.
      DN=0.
      BN=1.
C*
   34 CALL DEFQ(0,0,0,IER)
      IF(IER) GO TO 999
C*
C*  SCAN FOR POSSIBLE NEGATIVES AT JET EDGE--(UEDGE=0)
C*
      IF(UEDGE.NE.0.) GO TO 35
      DO 3391 L=1,NPD
      IF(VAR(L).LT.0.) GO TO 3392
 3391 CONTINUE
      GO TO 35
 3392 LK=L-1
      LL=L
      SLOPE=(VAR(NPD)-VAR(LK))/(SM(NPD)-SM(LK))
      NPD1=NPD-1
      DO 3393 L=LL,NPD1
 3393 VAR(L)=VAR(L-1)+SLOPE*(SM(L)-SM(L-1))
C*
   35 CALL MOVE(1,VAR,ED,NPD,1)
      CALL MOVE(2,DVAR,DIKE,NPD,1,BE,DSTOR(601),NPU,1)
      IF(DPRIN) CALL TABPRT(NAM(5),ED ,NPD,10,0)
C*
C*  TEST FOR SPECIES DEQS
C*
 7010 IF(.NOT.DIFF) GO TO 50
      CALL SETM(2,0.,GM,NPU,DL,NPU)
C*
C*  SOLVE SPECIES EQUATIONS// AIR MOLE FRACTION
C*  COMPUTED BY DIFFERENCE(COMPONENT 1)
C*
 7020 DO 7500 LL=2,NC
      NM1=NPU
      CALL MOVE(1,ALXU(1,LL),VAR,NPU,1)
      DO 7100 L=2,NPU
      RAD=1.
      IF(AXI) RAD=Y(L)**2
      RD=RHO(L)*UK(L)
      BE(L)=MUEFF(L)/SCM(LL)*RD*RAD
 7100 GM(L)=0.
      BE(1)=0.
      IF(.NOT.AXI) BE(1)=MUEFF(1)*RHO(1)*UO(1)/SCM(LL)
C*
```

```
C*  BOUNDARY CONDITIONS
C*
      AN=1.
      BN=0.
 7102 DN=ALEDGE(IL)
C*
 7110 CALL DFEQ(1,0,0.IER)
      IF(IER) GO TO 999
C*
C*  MOVE NEW MOLE FRACTIONS,ETC-INTERP. UPSTRM DALXU
C*
      DO 7115 L=1,NPU
      IF( VAR(L).LT.0. ) VAR(L)=0.
 7115 DALXU(L,LL)=YDF(SM1(L),SM,DVAR,1,NPD)
      CALL MOVE(1,VAR,ALX(1,LL),NPD,1)
C*
 7500 CONTINUE
C*
C*  COMPUTE AIR MOLE FRACTION AND UPSTREAM DERIVATIVE
C*
      DO 7749 L=1,NPD
      ALX(L,1)=1.
      DALXU(L,1)=0.
      DO 7750 LL=2,NC
      ALX(L,1)=ALX(L,1)-ALX(L,LL)
      DALXU(L,1)=DALXU(L,1)-DALXU(L,LL)
 7750 CONTINUE
 7749 CONTINUE
C*
      IF(.NOT.DPRIN) GO TO 50
      DO 5092 LL=1,NC
      CALL TABPRT(CNAME(LL),ALX(1,LL),NPD,10,0)
 5092 CONTINUE
C*  TEST FOR HEAT TRANSFER EFFECTS
C*
C*
   50 IF(.NOT. QJET) GO TO 6554
C****  SOLUTION OF ENERGY EQUATION
C*
   51 NM1=NPU
      CALL SETM(1,0,.DL,NPU,GM,NPU)
      IF(.NOT.TURBJ) GO TO 54
C*
C*
C*  COMPUTE D(DTKE/DY)/DY
C*
C*  COMPUTE TERMS ENTERING INTO SOURCE
C*
C*
      DO 52 L=1,NPU
```

```
      BE(L)=C3*DSTOR(L+600)*YOF(SM1(L),SM,DTKE,1,NPD)                   JTSTP451
   52 CONTINUE                                                          JTSTP452
      NPM=NPU-1                                                         JTSTP453
   53 CALL DERIV(SM1,BE,GM,2,NPM)                                       JTSTP454
C*                                                                      JTSTP455
   54 DO 55 L=2,NPU                                                     JTSTP456
   57 RAD=1.                                                            JTSTP457
      IF(AXI) RAD=Y(L)**2                                               JTSTP458
      RD=RHO(L)*UK(L)                                                   JTSTP459
      AL(L)=AL(L)/CP(L)                                                 JTSTP460
      BE(L)=KCP(L)*CP(L)*RD*RAD                                         JTSTP461
      DUDYSQ=DUDY(L)*DUDY(L)                                            JTSTP462
      GM(L)=GM(L)/CP(L)+C4*RD*RAD*MUEFF(L)*DUDYSQ/CP(L)                 JTSTP463
     * -(YOF(SM1(L),SM,ED,1,NPD)-EU(L))/(DX*CP(L))*EJET/TJET            JTSTP464
   58 IF(.NOT. DIFF) GO TO 55                                           JTSTP465
      ETA(L)=TJET*CND(1)*MUEFF(L)*RD*RAD/CP(L)*SUMCPD(L)                JTSTP466
   55 CONTINUE                                                          JTSTP467
      BE(1)=0.                                                          JTSTP468
      IF(.NOT. AXI)  BE(1)=KCP(1)*CP(1)*RHO(1)*UK(1)                    JTSTP469
C*                                                                      JTSTP470
      CALL MOVE(1,THU,VAR,NPU,1)                                        JTSTP471
      DN=THEDGE                                                         JTSTP472
      IF(.NOT.MIX) GO TO 56                                             JTSTP473
C*                                                                      JTSTP474
C*  B.C. FOR CONFINED MIXER                                             JTSTP475
C*                                                                      JTSTP476
      IF(UPSTRM) GO TO 56                                               JTSTP477
      AN=0.                                                             JTSTP478
      BN=1.                                                             JTSTP479
      DN=0.                                                             JTSTP480
   56 CALL DFEQ(0,0,0,IER)                                              JTSTP481
      IF(IER) GO TO 999                                                 JTSTP482
      IF(DIFF) CALL SETM(1,0.,ETA,200)                                  JTSTP483
C*                                                                      JTSTP484
      IF(UEDGE .NE. 0.) GO TO 60                                        JTSTP485
      DO 6691 L=1,NPD                                                   JTSTP486
      IF(VAR(L).LT. 0.) GO TO 6692                                      JTSTP487
 6691 CONTINUE                                                          JTSTP488
      GO TO 60                                                          JTSTP489
 6692 LK=L-1                                                            JTSTP490
      LL=L                                                              JTSTP491
      SLOPE=(VAR(NPD)-VAR(LK))/(SM(NPD)-SM(LK))                         JTSTP492
      NPD1=NPD-1                                                        JTSTP493
      DO 6693 L=LL,NPD1                                                 JTSTP494
 6693 VAR(L)=VAR(L-1)+SLOPE*(SM(L)-SM(L-1))                             JTSTP495
   60 CALL MOVE(1,VAR,THD,NPD,1)                                        JTSTP496
      IF(DPRIN) CALL TABPRT(NAM(6),THD,NPD,10,0)                        JTSTP497
C*                                                                      JTSTP498
C*                                                                      JTSTP499
C*  COANNULAR PROBLEM-- IF .NOT. MERGE.  COMPUTE U0                     JTSTP500
```

```
C*
 6554 IF(.NOT. TWO) GO TO 6555
      IF(MERGE.OR. (.NOT. MIX)) GO TO 6555
      IF(MERGE) GO TO 6555
      U0=SQRT(USV0**2-144.*GC*RG*TJET*THSV0*(P2-P1)/P1)
 6555 CONTINUE
C*
C*
C*
C*
C*  LOCATE EDGE OF JET--- ADD POINT IF NECESSARY
C*
   86 CALL JTEDGE(X,YJETE,SFEDGE,ADDP)
C*
   87 CONTINUE
C*
C*  CONFINED MIXER-CHECK FOR CONVERGENCE
C*  OF PRESSURE ITERATION
C*
C*
      IF(.NOT.MIX) GO TO 80
      CALL AITER2
      IF(ADDP .AND. (.NOT. UPSTRM))  ADDP=.FALSE.
      IF(ERR) RETURN
      IF(CVG) GO TO 80
      NM1=NPU
      CALL MOVE(1,YU,Y,NPU,1)
      ICYCLE=.TRUE.
      IF(.NOT. TWO) GO TO 9966
C*
C*  RESTORE ON ITERATION
C*
      U0=USV0
      TH0=THSV0
      GO TO 9966
C*
C*  IF UPSTRM=F, SET SM(NM)= PSID FOR CONFINED MIXER
C*  NO MESH POINTS WILL BE ADDED AFTER THIS POINT
C*
   80 IF (MIX .AND. (.NOT. UPSTRM))  SM(NM)=PSID
C*
C*  MOVE DOWNSTREAM CO-ORDINATES TO UPSTREAM TABLE
C*
C*
      CALL MOVE(1,SM,SM1,NM,1)
      ICYCLE=.FALSE.
C*
C*  TWO- JET LOGIC TO TEST FOR INTERACTION OF INNER AND OUTER JETS
C*
      IF((.NOT. TWO)  .OR.  (.NOT. MERGE)  .OR.  MERSTP) GO TO 776
```

```
C*
C*  JETS HAVE MERGED
C*
      MER=.TRUE.
      MERSTP=.TRUE.
      XMRG=XMERGE
      GO TO 102
C*
C*
C*  TEST FOR DISAPPEARANCE OF SUPERSONIC CORE IF JET IS SUPERSONIC
C*
  776 IF(SUBSON)  GO TO 777
      TCL=TJET*THD(1)
      VCL=VJET*UD(1)
      VSONC = SQRT(GMC(1)*GC*RG*TCL)
      IF(VCL.GE.VSONC) GO TO 777
C*
C*  SUPERSONIC CORE HAS JUST DISAPPEARED
C*.
      SUPC=.TRUE.
      SUPSTP=.TRUE.
C*
C*  FLAG NOW SUBSONIC JET
C*
      SUBSON=.TRUE.
      XSUP=X
      GO TO 102
C*
C*
C*  ** SEQUENCE OF TESTS FOR DISAPPEARANCE OF THE
C*  ** POTENTIAL CORE OR THE LAST STEP
  777 IF(LAST) GO TO 220
      IF( SCLD ) GO TO 778
      IF( UD(2).EQ.UD(3)) GO TO 310
      GO TO 779
  778 IF( ALX(1,NC).LE.ALXLIM ) GO TO 310
  779 IF( CORSTP ) GO TO 310
C*
C*  CORE HAS JUST DISAPPEARED
C*
      CORE=.TRUE.
  101 CORSTP=.TRUE.
      NREG=2
      XLC=X
      XCORE=X
  102 CALL XSIZE(DX,X,REFL,NREG,LAST)
      GO TO 500
C*
C*
C*  TEST FOR DISAPPEARANCE OF CORE
```

```
C*                                                                      JTSTP601
  220 LAST=.FALSE.                                                      JTSTP602
      IF( SCLD ) GO TO 224                                              JTSTP603
      IF((UD(2).EQ.UD(3)) .OR.  CORSTP) GO TO 500                       JTSTP604
      GO TO 101                                                         JTSTP605
C*                                                                      JTSTP606
C*  CORE HAS JUST DISAPPEARED-SET CORSTP=T.                             JTSTP607
C*                                                                      JTSTP608
  224 IF( ALX(1,NC).LE.ALXLIM .OR. CORSTP ) GO TO 500                   JTSTP609
      GO TO 101                                                         JTSTP610
C*                                                                      JTSTP611
C*                                                                      JTSTP612
C*  TEST FOR NO. OF MESH POINTS.GT.NM                                   JTSTP613
C*                                                                      JTSTP614
  310 IF(NPD.GT. NMAX) CALL MSHCUT(NREG,SM,NPD)                         JTSTP615
C*                                                                      JTSTP616
C*  ADJUST X-STEP SIZE                                                  JTSTP617
C*                                                                      JTSTP618
  320 CALL XSIZE(DX,X,REFL,NREG,LAST)                                   JTSTP619
      CALL DTEST                                                        JTSTP620
      IF(ERR) GO TO 500                                                 JTSTP621
C*                                                                      JTSTP622
      CALL MOVE(3,UD,UU,NM1,1,ED,EU,NM1,1,THD,THU,NM1,1)                JTSTP623
      NPU=NM1                                                           JTSTP624
      IF(.NOT.DIFF) GO TO 8                                             JTSTP625
      DO 7700 LL=1,NC                                                   JTSTP626
      CALL MOVE(1,ALX(1,LL),ALXU(1,LL),NM1,1)                           JTSTP627
 7700 CONTINUE                                                          JTSTP628
      GO TO 8                                                           JTSTP629
C*                                                                      JTSTP630
C*  ERROR RETURN                                                        JTSTP631
C*                                                                      JTSTP632
  999 ERR=.TRUE.                                                        JTSTP633
C*                                                                      JTSTP634
  500 CONTINUE                                                          JTSTP635
      NPU=NM1                                                           JTSTP636
 1000 CALL MOVE(4,UD,UU,NM1,1,ED,EU,NM1,1,THD,THU,NM1,1,                JTSTP637
     * SM1,PSI,NM1,1)                                                   JTSTP638
      IF(.NOT.DIFF) GO TO 7740                                          JTSTP639
      DO 7730 LL=1,NC                                                   JTSTP640
      CALL MOVE(1,ALX(1,LL),ALXU(1,LL),NM1,1)                           JTSTP641
 7730 CONTINUE                                                          JTSTP642
 7740 ASSIGN 1001 TO LGOP                                               JTSTP643
      GO TO 10                                                          JTSTP644
C*                                                                      JTSTP645
 1001 IF(PM(10).NE. 0.)  CALL TABPRT(NAM(3),XLN,NPD,10,0)               JTSTP646
      RETURN                                                            JTSTP647
      END                                                               JTSTP648
```

```
CKEBLOK                                                                 KEBLOK01
      BLOCK  DATA                                                       KEBLOK02
C                                                                       KEBLOK03
      COMMON /PSEQX / HQC,HUM,CO2AIR,MAIR,FS,FUELMW                     KEBLOK04
      REAL                          MAIR                                KEBLOK05
      COMMON /COIREM/ YTOL,Y0,DYDX,CTRMAX                               KEBLOK06
      COMMON /CKINET/ TIME,DTIME,NTSTEP,DBET,XMOLWO,CRC,NREAC,          KEBLOK07
     *                RC1(9),RC2(9),RC3(9),TBE(55),RCON(5),TK,ENTRY1    KEBLOK08
      LOGICAL         CRC,ENTRY1                                        KEBLOK09
      COMMON /KININS/ XMWUN,XMWC,TCONST,CONER,XNOI                      KEBLOK10
      LOGICAL      TCONST,CONER                                         KEBLOK11
      COMMON /CCOCO2/ CORATE,XCOI                                       KEBLOK12
      LOGICAL         CORATE                                            KEBLOK13
C                                                                       KEBLOK14
      DATA HQC,HUM,CO2AIR,MAIR/2.,0.,.0003,28.9666/                     KEBLOK15
      DATA YTOL,CTRMAX/5.E-3,100./                                      KEBLOK16
      DATA XMOLWO/O377777777777/                                        KEBLOK17
      DATA CRC,NREAC/F,5/                                               KEBLOK18
      DATA TCONST,CONER,XNOI/F,T,0./                                    KEBLOK19
      DATA CORATE/T/                                                    KEBLOK20
      DATA RC1/1.E+19,3.839E+13,5.3E+15,8.15E+18,1.E+15,7.E+13,         KEBLOK21
     * 2*6.E+12,2.3E+13/                                                KEBLOK22
      DATA RC2/-1.,.1095,0.,-1.22,5*0./                                 KEBLOK23
      DATA RC3/0.,-10594.,-2780.,0.,-1300.,4*0./                        KEBLOK24
      DATA TBE/2*1.07,3*4.5,18.,4.5,8.,3.6,1.,4.5,33*1.,2*1.,5.,2*2.,   KEBLOK25
     * 32.5,2.,7.5,2.,1.,2./                                            KEBLOK26
      DATA NTSTEP,DBET/ 10,.01 /                                        KEBLOK27
C                                                                       KEBLOK28
      END                                                               KEBLOK29
```

```
CKINET        HYDROCARBON KINETICS ROUTINE
      SUBROUTINE KINET(P,H,FOA1,DT)
C
      CHEM(X,Y)= (X*((X+1.-BETA0)*EXP(Y)-(X-1.+BETA0)))/
     *           ((X+1.-BETA0)*EXP(Y)+(X-1.+BETA0))
      LOGICAL RATECO
      LOGICAL  MOLIN
      COMMON /CBITS / BITS,BLANK
      COMMON /PSEQ  / FOA,BETA,TP,X(16),DHQDMW,TEQ,BEQ,XMWT,HNEQ
      EQUIVALENCE (X1,X(1)),(X2,X(2)),(X3,X(3)),(X4,X(4)),
     *            (X5,X(5)),(X6,X(6)),(X7,X(7)),(X8,X(8)),
     *            (X9,X(9)),(X10,X(10)),(X11,X(11))
      COMMON /PROPR / PR,HHR,TR,FCR,RHOR,RR,WMWTR
      COMMON /CKINET/ TIME,DTIME,NTSTEP,DDFT,XMOLW0,CRC,NREAC,
     *                RC1(9),RC2(9),RC3(9),TBE(11,5),RCON(5),
     *                TK,ENTRY1
      LOGICAL         CRC,ENTRY1
      EQUIVALENCE (RK1,RCON(1)),(RK2,RCON(2)),(RK3,RCON(3)),
     *            (RK4,RCON(4)),(RK5,RCON(5))
      COMMON /KININS/ XMWUN,XMWC,TCONST,CONER,XNOI
      LOGICAL                   TCONST,CONER
      LOGICAL FIRST, LAST
      COMMON /CQIREM/ YTOL,DUM(3)
      COMMON /CPRINT/ PDUM(20)
      COMMON /INMOLF/ XIN(12)
      COMMON /CRATE / RATE
      COMMON /GHSC  / FF(25),HZ(25),SR(25),CPZ(25),DCPR(25)
      COMMON /CEQKIN/ BK,A1K,DELHNO,DEN,XMWEQ,XND,XCO
      COMMON /CCOCO2/ CORATE,XCOI
      LOGICAL         CORATE
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
      LOGICAL         ERR,ERRMAJ,INERR,PRERR
      COMMON /PSEQX / HOC,HUM,CO2AIR,MAIR,FS,FUELMW
      REAL                           MAIR
      COMMON /CPSEQ2/ SPM,SPMU,SPMC
      COMMON /DPBETA/ BETADP
      DOUBLE PRECISION BETADP
      COMMON /SNMW  / ALSP(150), WMTC(75)
      COMMON /COLIMT/ XCOLIM
      DIMENSION QV(8)
      DIMENSION XSAV(12)
C
      DATA C2/.01603286/
      DATA R0/1.98596/
      NAMELIST /A/ ER1,ERSAV,BETA,BETS,ENTRY1,IK,TIME,
     * DTIME,TP,XMW1,XMWT,XMWC,XMWUN,X,A1,RATE,XMWEQ,XND
     * ,DHQDMW,DELH,HHR,RCON,BETA0,BK,XMWEQ,DELHNO ,CPM
     * ,BEQ,TEQ
     *,QV,TP1,HH,HH1,ERRT,BETADP
C
```

```
C     **REACTIONS**     (1)    H   +OH  +M    =   H2O+M                              KINET051
C                       (2)    H   +H   +M    =   H2 +M                              KINET052
C                       (3)    H   +O   +M    =   OH +M                              KINET053
C                       (4)    O   +O   +M    =   O2 +M                              KINET054
C                       (5)    H   +O2  +M    =   HO2+M                              KINET055
C                       (6)    H   +HO2       =   OH +OH                             KINET056
C                       (7)    OH  +HO2       =   H2O+O2                             KINET057
C                       (8)    O   +HO2       =   OH +O2                             KINET058
C                       (9)    H   +HO2       =   H2 +O2                             KINET059
C                                                                                    KINET060
C                                                                                    KINET061
C                                                                                    KINET062
       IADD  = 0                                                                     KINET063
       IF( CONER .AND. (.NOT.ENTRY1) ) GO TO 20                                      KINET064
       IF( .NOT.ENTRY1 ) GO TO 1                                                     KINET065
       FUELMW= 12.01+1.008*HOC                                                       KINET066
       FS    = FUELMW/(1.+.25*HOC)*.209495/MAIR                                      KINET067
     1 FOA   = FOA1                                                                  KINET068
       ER1   = FOA/FS                                                                KINET069
       HH    = H                                                                     KINET070
       IF( ERSAV.EQ.0. ) GO TO 12                                                    KINET071
C                                                                                    KINET072
C     ADJUST BETA, COMPOSITIONS FOR ADDED AIR                                        KINET073
C                                                                                    KINET074
       F1    = ERSAV*FS                                                              KINET075
       F2    = FOA1                                                                  KINET076
       FB1   = F1/(1.+F1+HUM)                                                        KINET077
       FB2   = F2/(1.+F2+HUM)                                                        KINET078
       DEL   = FB1/FB2-1.                                                            KINET079
       XMSAV = XMW1                                                                  KINET080
       TERMW = DEL*(1./MAIR+HUM/WMTC(6))/(1.+HUM)                                    KINET081
       XMW1  = (1.+DEL)/(1./XMSAV+TERMW)                                             KINET082
       TERMW1= XMW1/(1.+DEL)                                                         KINET083
       TERMP1= DEL/(1.+HUM)                                                          KINET084
       TERMP2= XMW1/(XMSAV*(1.+DEL))                                                 KINET085
       CALL FMPYC(1,TERMP2,X,X,3)                                                    KINET086
       X5    = TERMP2*X5                                                             KINET087
       X7    = TERMP2*X7                                                             KINET088
       X11   = TERMP2*X11                                                            KINET089
       X4    = TERMW1*(X4/XMSAV+TERMP1*.209495/MAIR)                                 KINET090
       X6    = TERMW1*(X6/XMSAV+TERMP1*HUM/WMTC(6))                                  KINET091
       X8    = TERMW1*(X8/XMSAV+TERMP1*CO2AIR/MAIR)                                  KINET092
       X9    = TERMW1*(X9/XMSAV+TERMP1*.780881/MAIR)                                 KINET093
       X10   = TERMW1*(X10/XMSAV+TERMP1*(.009624-CO2AIR)/MAIR)                       KINET094
       TMOLES= 0.                                                                    KINET095
       DO 5 I=1,11                                                                   KINET096
     5 TMOLES= TMOLES+X(I)                                                           KINET097
       RADDM = 1./TMOLES                                                             KINET098
       CALL FMPYC(1,RADDM,X,X,12)                                                    KINET099
       XNO   = X(11)                                                                 KINET100
```

```
      XCO    = X7
      IADD   = 1
C
C
C     IF FIRST ENTRY-- INITIALIZE COMPOSITIONS AT XMOLWO
C
   12 IF( .NOT.ENTRY1 ) GO TO 20
      DHODMW= 300.
      XNO    = XNOI
      XCO    = XCOI
      TIME   = 0.
      FRSAV  = 0.
      MOLIN  = .FALSE.
      CALL SETM(1,0.,X,11)
      X11    = XNOI*X3
      X7     = XCOI*X3
      IF( XMOLWO.NE.BITS ) GO TO 15
      MOLIN  = .TRUE.
      XMOLWO= 0.
      DO 13 K=1,12
   13 XMOLWO= XMOLWO+XIN(K)*WMTC(K)
      XNO    = XIN(11)
      XCO    = XIN(7)
      CALL MOVE(1,XIN,X,12,1)
   15 CALL EQKIN( P,HH,.FALSE. )
      BETA   = (1./XMOLWO-1./XMWUN)*DEN
      BETAO  = BETA
   16 BETADP= BETA
      IF(MOLIN) GO TO 17
      X7     = XCO
      X11    = XNO
      CALL PSEQ2(FOA,HUM,HQC,TP,BETADP,CORATE,.TRUE.,X)
      GO TO 18
   17 CALL MOVE(1,XIN,X,12,1)
   18 XNO    = X11
      XCO    = X7
      XMWI   = XMOLWO
      XMWT   = XMOLWO
      RHO    = 144.*XMWT*P/(1545.32*TP)
      IF( TCONST ) HNEQ=BITS
      ENTRY1= .FALSE.
      RETURN
C
C     INTEGRATE RATE EQUATIONS OVER GIVEN TIME STEP
C
C
C     ESTIMATE NUMBER OF TIME STEPS AT ER1
C
   20 IF( NTSTEP.EQ.0 ) GO TO 291
      DTIME = DT/FLOAT(NTSTEP)
```

```
      IF( BK.EQ.0. ) IK=1                                              KINET151
      GO TO 223                                                        KINET152
  291 IF( BETA.GE.1. ) GO TO 222                                       KINET153
      IF( BK.NE.0. ) GO TO 21                                          KINET154
      IK     = 0                                                       KINET155
      ASSIGN 201 TO LGO                                                KINET156
      GO TO 24                                                         KINET157
  201 DTIMAX= DBET/A1                                                  KINET158
      IK     = 1                                                       KINET159
      GO TO 222                                                        KINET160
   21 ASSIGN 22 TO LGO                                                 KINET161
      GO TO 24                                                         KINET162
   22 DTIMAX= DBET/(A1*(1.-BETA)**2-BK)                                KINET163
  222 DTIME = AMIN1( DT,DTIMAX )                                       KINET164
  223 TIME2 = TIME+DT                                                  KINET165
      ASSIGN 32 TO LGO                                                 KINET166
      LAST  = .FALSE.                                                  KINET167
      TIME  = TIME+DTIME                                               KINET168
      IF( TIME.LT.TIME2 ) GO TO 241                                    KINET169
      LAST  = .TRUE.                                                   KINET170
  241 CALL MOVE(1,X,XSAV,12,1)                                         KINET171
      IF( TCONST ) GO TO 245                                           KINET172
      CALL EQKIN( P,HH,.FALSE. )                                       KINET173
      XCOLIM= X7                                                       KINET174
      RHO   = 144.*XMW1*P/(1545.32*TP)                                 KINET175
      IF( ER1.LT.PDUM(6).AND.ER1.GE.PDUM(5) ) WRITE (6,A)              KINET176
      RATECO= CORATE                                                   KINET177
  245 CALL MOVE(1,XSAV,X,12,1)                                         KINET178
      IF( CONER ) GO TO 24                                             KINET179
      BETA  = (1./XMW1-1./XMWUN)*DEN                                   KINET180
      IF( BETA.GE.BEQ ) BETA=BEQ                                       KINET181
      IF( TP.GE.TEQ ) TP=TEQ                                           KINET182
      BETAO = BETA                                                     KINET183
      IF( IADD.EQ.0 ) GO TO 24                                         KINET184
C                                                                      KINET185
C*    RECALCULATE PSEUDO-EQUILIBRIUM AFTER DILUTION                    KINET186
C                                                                      KINET187
      BETADP= BETA                                                     KINET188
      QV(1) = 0.                                                       KINET189
      TP1   = TP                                                       KINET190
      XJP   = .005*TP                                                  KINET191
 2451 CALL PSEQ2(FOA,HUM,HQC,TP1,BETADP,RATECO,.TRUE.,X)               KINET192
      HH1   = 0.                                                       KINET193
      DO 2452 K=1,11                                                   KINET194
 2452 HH1   = HH1+X(K)*HZ(K)                                           KINET195
      HH1   = RO*TP1*HH1/XMW1                                          KINET196
      ERRT  = HH-HH1                                                   KINET197
      CALL QIKEM(TP1,ERRT,XJP,QV)                                      KINET198
      IF( QV(1).GE.80. ) WRITE (6,36) TP,TP1,ERRT,HH,HH1,X             KINET199
      IF( QV(1).NE.0. ) GO TO 2451                                     KINET200
```

```
      TP     = TP1                                                      KINET201
      IADD   = 0                                                        KINET202
C                                                                       KINET203
C                                                                       KINET204
C     IF CORATE=T,  INTEGRATE CO RATE EQUATION                          KINET205
C                                                                       KINET206
   24 IF( .NOT.RATECO ) GO TO 2460                                      KINET207
      XCOSAV= X7                                                        KINET208
      CALL COCO2B( P,DTIME )                                            KINET209
      XCO    = X7                                                       KINET210
      DXCO   = XCO-XCOSAV                                               KINET211
      X6     = X6+DXCO                                                  KINET212
      X3     = X3-DXCO                                                  KINET213
C                                                                       KINET214
C     CALCULATE FULL EQUILIBRIUM WITH FIXED CO,NO                       KINET215
C                                                                       KINET216
      CALL MOVE(1,X,XSAV,12,1)                                          KINET217
      CALL EQKIN( P,HH, .TRUE. )                                        KINET218
      CALL MOVE(1,XSAV,X,12,1)                                          KINET219
 2460 IF(ER1.LT.PDUM(6).AND.ER1.GE.PDUM(5)) WRITE (6,A)                 KINET220
      IF( BETA.EQ.BEQ ) GO TO 40                                        KINET221
      TK     = TP/1.8                                                   KINET222
      RHOCGS= C2*RHO                                                    KINET223
   25 CALL RATCON(P)                                                    KINET224
   30 RATE   = -RHOCGS**2/ XMW1**3*(RK1*X1*X5+RK2*X1**2+RK3*X1*X2       KINET225
     * +RK4*X2**2+RK5*X1*X4)                                            KINET226
      IF( RATE.EQ.0. ) GO TO 33                                         KINET227
      A1     = RATE/((1./XMW1-1./XMWC)**2*DEN)                          KINET228
      IF( BK.NE.0. ) GO TO 31                                           KINET229
      IF( IK.EQ.0 ) GO TO 301                                           KINET230
C                                                                       KINET231
C     LIMITING BETA AS BK GOES TO 0                                     KINET232
      BETA   = 1.-(1.-BETA0)/(1.+(1.-BETA0)*A1*DTIME)                   KINET233
      GO TO 33                                                          KINET234
C  31 AA     = SQRT(BK/A1)                                              KINET235
C     AAA    = 2.*SQRT(BK*A1)                                           KINET236
C************* MODS FOR TEST DECK  *************************************KINET237
   31 AA     = 1.-BEQ                                                   KINET238
      AAA    = 2.*SQRT((A1*AA)**2)                                      KINET239
C**********************************************************************KINET240
  301 GO TO LGO , (22,32,201)                                           KINET241
   32 AAA    = AAA*DTIME                                                KINET242
      BETA   = 1.-CHEM(AA,AAA)                                          KINET243
   33 XMWT   = 1./(BETA/DEN+1./XMWUN)                                   KINET244
      IF( BETA.GE.BEQ ) BETA=BEQ                                        KINET245
      IF( BETA.EQ.BEQ ) XMWT=XMWEQ                                      KINET246
      BETA0 = BETA                                                      KINET247
      IF( TCONST ) GO TO 40                                             KINET248
C                                                                       KINET249
C                                                                       KINET250
```

```
C     ITERATE FOR TEMPERATURE
C
      BETADP= BETA
      QV(1) = 0.
      TP1   = TP
      XJP   = .01*TP
   34 CALL PSEQ2(FOA,HUM,HOC,TP1,BETADP,RATECO,.TRUE.,X)
      HH1   = 0.
      DO 35 K=1,11
   35 HH1   = HH1+X(K)*HZ(K)
      HH1   = R0*TP1*HH1/XMWT
      ERRT  = HH-HH1
      CALL DIREM(TP1,ERRT,XJP,QV)
      IF( QV(1).GE.80. ) WRITE (6,36) TP,TP1,ERRT,HH,HH1,X
   36 FORMAT(/2X,39HKINET T ITERATION  TP,TP1,ERRT,HH,HH1,X/
     * 1X,5E16.8/1X,6E16.8/1X,6E16.8)
      IF( QV(1).NE.0. ) GO TO 34
      TP    = TP1
C
   40 XNO   = X11
      XMW1  = XMWT
      RHO   = 144.*XMWT*P/(1545.32*TP)
C
C     PREDICT NO AT END OF TIME STEP
  401 XNOSAV= XNO
C
      CALL NOX2B(P,DTIME,XNO )
C
C     ADJUST MOLE FRACTIONS FOR NO-- N ATOMS NOT USED
      DXNO  = XNO-XNOSAV
      X4    = X4-.5*DXNO
      X9    = X9-.5*DXNO
      X11   = XNO
   99 IF( LAST ) GO TO 100
      TIME  = TIME+DTIME
      IF( TIME.LT.TIME2 ) GO TO 24
      LAST  = .TRUE.
      TIME  = AMIN1( TIME,TIME2 )
      GO TO 24
  100 ERSAV = ER1
      TIME  = TIME2
      RETURN
      END
```

```
CKINPP       READ INPUT DATA AND INITIALIZE
C            SPECIAL VERSION - GE-AEG
C      8/23/73  ** MODIFICATIONS FOR GCKP-1 **
       SUBROUTINE KINP
C
C     INPUT CAN BE ACCEPTED IN  (1) INTERNAL (CGS) UNITS, (2) FPS UNITS,
C      (3) SI UNITS
C
C     THE FOLLOWING UNITS ARE USED INTERNALLY
C     * DISTANCE          CM                  *
C     * AREA              CM**2               *
C     * MASS FLOW RATE    GM/SEC              *
C     * PRESSURE          ATM                 *
C     * TIME              SEC                 *
C     * VELOCITY          CM/SEC              *
C     * DENSITY           GM/CC               *
C     * TEMPERATURE       DEG K               *
C     * CONCENTRATION     MOLE(I)/MASS        *
C
C     INTERNAL CORRESPONDENCE
C     * DVAR    -   DEPENDENT VARIABLE        *
C     * IVAR    -   INDEPENDENT VARIABLE      *
C     * AVAR    -   ASSIGNED VARIABLE         *
C
C     THE FOLLOWING LOGICAL TAPE UNITS ARE REQUIRED
C     * LTHM (4)     -   FOR THERMODYNAMIC DATA                    *
C     * LDAT (43)    -   FOR TEMPORARY STORAGE OF DATA CARDS       *
C     * 03           -   FOR OPTIONAL BINARY OUTPUT FILE           *
C     LOGICAL TAPE UNIT ASSIGNMENTS ARE SPECIFIED IN 'NAMBLK'
C
      DOUBLE PRECISION DSP,DSPP,DSPNM,DAISP
C
      LOGICAL ALLMI,CONC,DRUGO,ELIM,FVSTEP,EXCHR,MOLFF,MMHG
      LOGICAL COMBUS,RHOCON,SHOCK,TCON
      LOGICAL NEWPRT,NEXT,BRIEF
C
      REAL MDOT,IVAR,M,MW,N,LSUBM,MIXMW,M2,NEW
C
      DIMENSION ISS(25),IBR(3),CXIB(40),CAIB(40),APRINT(50),THMC(7,2)
      DIMENSION SP(2,4),DSP(4),SPP(2,3),DSPP(3),SPNM(2,27),DSPNM(27)
      DIMENSION SPT(2),LMT(4),SUBS(4),C(25),CX(4)
      DIMENSION CUA(2),FUA(2),SUA(2),CUP1(2),CUP2(2),FUP(2),SUP(2)
      DIMENSION  HEAD(10)
C
      COMMON/LTUS/LTHM,LDAT
      COMMON/OPTS/VERSI,TIMEV,VERSA,AREAV,ELIM,TCON,RHOCON,IPRCOD
      COMMON/COND/DVAR,AREA,MDOT,P,IVAR,V,RHO,T,SIGMA(25),LS,LSP3,NEXT
      COMMON/REAC/LSR(4,30),XX(30),RATE(30),LKEQ(30),DLKEQ(30),MM(30),LR
      COMMON/RRAT/A(30),N(30),EACT(30),B(30),M(25,30),ALLMI
      COMMON/AFUN/CN(4),ITPSZ,LSUBM,ETA,D,VISC,BETA
```

```
      COMMON/SPEC/SNAM(2,30),MW(25),W(25),STOIC(25,30),OMEGA(25,30)     KINPP051
      COMMON/SINT/HMIN,HI,HN,HNP1,HMAX,NH,AVH,EMAX,ERRN,ICV,KOUNT,FRRP  KINPP052
      COMMON/TCOF/TC(7,2,25),TLOW,TMID,THI                              KINPP053
      COMMON/PRIN/PRINT(50),NP,CEND,EVSTEP                              KINPP054
      COMMON/XVSA/XTB(40),ATB(40),NT,XU,AU(2),CX3,CX2,CX1,CX0           KINPP055
      COMMON/XVSA/BRIEF,TIMLMT                                          KINPP056
      COMMON/SNMW/DALSP(75),ALMW(75)                                    KINPP057
      COMMON/KOUT/TITLE(20),UNITT,UNITO,CONC,EXCHR,DELH(30),FPS,SI,DBUGOKINPP058
      COMMON/GHSC/GRT(25),HRT(25),SR(25),CPR(25),DCPR(25)               KINPP059
      COMMON/PORF/PK(28),QK(28),RK(28),E(28)                            KINPP060
      COMMON/SKIP/NEGL(25),T1,T2,IT                                     KINPP061
      COMMON/NECC/RR,MIXMW,M2,GAMMA,TCPR,R                              KINPP062
      COMMON/MISC/TI,PP,CPRO,HPO,ENN,SUMN,ENNL,LLMT(15),BO(15)          KINPP063
      COMMON/INDX/TP,HP,NLM,NS,IQ1,CONVG,KMAT,IMAT                      KINPP064
      COMMON / LSLRSV / LSSV, LRSV                                      KINPP065
      COMMON /CINSAV/ HMINZ,HMAXZ,HINTZ,FMAXZ,PRINTZ(50),NPRNTZ,        KINPP066
     *                EVSTPZ,DBUGOZ                                     KINPP067
      LOGICAL         EVSTPZ,DBUGOZ                                     KINPP068
      COMMON /STCTRL/ DUMST(4),FIRSTC,DUMST1(12)                        KINPP069
      LOGICAL                        FIRSTC                             KINPP070
C                                                                       KINPP071
      EQUIVALENCE (C,SIGMA),(SPNM,DSPNM),(DSP,SP),(DSPP,SPP),(SPT,SP)   KINPP072
      EQUIVALENCE (SPNM,SNAM(1,4)),(EEEM,SPNM(1,26)),(BLANK,SPNM(1,27)) KINPP073
      EQUIVALENCE (EX3,CX),(CEND,END)                                   KINPP074
C                                                                       KINPP075
      DATA CU,FU,SU/2HCM,2HFT,2HM /                                     KINPP076
      DATA CUA/4HCM**,1H2/,FUA/4HFT**,1H2/,SUA/4HM**2,1H /              KINPP077
      DATA CUP1/4HMMHG,1H /,CUP2/3HATM,1H /,FUP/4HLB/F,4HT**2/,SUP/4HN/MKINPP078
     **,2H*2/                                                           KINPP079
      DATA NEW,CHANGE,REPEAT/3HNEW,4HCHAN,4HREPE/                       KINPP080
      DATA TAPEND,CARDS/3HEND,4HCARD/                                   KINPP081
      DATA HEAD(1)/60H                   *****    GCKP=1 INPUT     *****KINPP082
     1           /                                                      KINPP083
C                                                                       KINPP084
      NAMELIST/PROB/HMIN,HMAX,HINT,EMAX, ALLM1,ELIM,CONC,EXCHR,         KINPP085
     * IPRCOD,ITPSZ, XTB,ATB,NTB, CX3,CX2,CX1,CX0, LSUBM,ETA,D,VISC,BETAKINPP086
     *, END,DELP, PRINT,NPRNTS,APRINT, EVSTEP,DBUGO,                    KINPP087
     * COMBUS,SHOCK,TCON,RHOCON,BRIEF,TIMLMT                            KINPP088
C                                                                       KINPP089
C                                                                       KINPP090
C                                                                       KINPP091
C                                                                       KINPP092
C     THERMODYNAMIC DATA WILL BE INPUT FROM BLOCK DATA =BLKTH=          KINPP093
C                                                                       KINPP094
      ACTION = NEW                                                      KINPP095
      GO TO 4                                                           KINPP096
C                                                                       KINPP097
      ENTRY RINP                                                        KINPP098
      NEXT = .FALSE.                                                    KINPP099
C                                                                       KINPP100
```

```
C
      IF (ACTION .NE. NEW) GO TO 9
C     SET STANDARD OPTIONS
    4 CONC = .TRUE.
      EXCHR = .FALSE.
      COMBUS = .FALSE.
      SHOCK = .FALSE.
      TCON = .FALSE.
      RHOCON = .FALSE.
      ELIM  = .FALSE.
      EVSTEP = .FALSE.
      DBUGO = .FALSE.
      EMAX = 0.0001
      ITPSZ = 5
      ALLMI = .FALSE.
C
C     M(I,J)      SET IN -BLKRDA
C
      BRIEF=.FALSE.
      TIMLMT=25.
C
C     INITIALIZE
      NEXT = .FALSE.
      HINT = 0.
C     NLM        SET IN -BLKEL-
C     NS       SET IN -BLKEL-
C     LS       SET IN -BLKRDA-      COMMON / SLRSV /
C     LR       SET IN -BLKRDA-      COMMON / SLRSV /
      NT = 0
      DO 6 I=1,40
      XTB(I) = 0.
      CXTB(I) = 0.
      ATB(I) = 0.
    6 CATB(I) = 0.
      UNCEND = 0.
      CEND = 0.
C
C     STOIC(I,J)      SET IN -BLKTH-
C
      LSUBM = 0.
      ETA = 0.
      D = 0.
      VISC = 0.
      BETA = 0.
      CX3 = 0.
      CX2 = 0.
      CX1 = 0.
      CX0 = 0.
      GO TO 14
C
```

```
    9 IF (ACTION .NE. CHANGE) GO TO 13                               KINPP151
C       REACTIONS AND REACTION RATE - -                              KINPP152
C       REACTION RATE VARIABLES ( A - N - FACT - LSR )  - -          KINPP153
C       SET IN -BLKRDA-                                              KINPP154
C                                                                    KINPP155
   13 IF (ACTION .EQ. REPEAT) GO TO 33                               KINPP156
C                                                                    KINPP157
   14 LSOLD =0                                                       KINPP158
      LROLD = 0                                                      KINPP159
C                                                                    KINPP160
C                                                                    KINPP161
C       SPECIES NAME AND MOLECULAR WEIGHT SET IN -BLCK-              KINPP162
C                                                                    KINPP163
C                                                                    KINPP164
C                                                                    KINPP165
C                                                                    KINPP166
   21 IF (ACTION .NE. NEW) GO TO 25                                  KINPP167
C       INERT SPECIES SET IN -BLCK-                                  KINPP168
C                                                                    KINPP169
C       SET INPUT SPECIES NAME EQUAL TO MASTER LIST NAME - -         KINPP170
C       SET COUNTERS ( LS - LR - LSOLD ) - - SET WM  AND  ISS        KINPP171
C                                                                    KINPP172
      LS = LSSV                                                      KINPP173
      LR = LRSV                                                      KINPP174
      LSOLD = LS                                                     KINPP175
C                                                                    KINPP176
      DO 23 II=1,LS                                                  KINPP177
      DSPNM(II) = DALSP(II)                                          KINPP178
      MW(II) = ALMW(II)                                              KINPP179
      ISS(II) = II                                                   KINPP180
   23 CONTINUE                                                       KINPP181
C                                                                    KINPP182
   25 IF (LS .EQ. LSOLD) GO TO 30                                    KINPP183
C                                                                    KINPP184
C     DETERMINE STOICHIOMETRIC COEFFICIENTS                          KINPP185
C                                                                    KINPP186
C       STOICHIOMETRIC COEFFICIENTS SET IN -BLKTH-                   KINPP187
C                                                                    KINPP188
   30 LRP = LROLD + 1                                                KINPP189
C                                                                    KINPP190
C     GET SPECIES ENTHALPY AT REFERENCE T                            KINPP191
      TREF = 298.15                                                  KINPP192
      CALL THRM (TREF,0.)                                            KINPP193
C                                                                    KINPP194
      TRAL = TREF*1.987165                                           KINPP195
      DO 32 J=LRP,LR                                                 KINPP196
      N1 = LSR(1,J)                                                  KINPP197
      N2 = LSR(2,J)                                                  KINPP198
      N3 = LSR(3,J)                                                  KINPP199
      N4 = LSR(4,J)                                                  KINPP200
```

```
C     COMPUTE HEAT OF REACTION
      DELH(J) = HRT(N3) - HRT(N2)
      IF (N1 .GT. 0) DELH(J) = DELH(J) - HRT(N1)
      IF (N4 .GT. 0) DELH(J) = DELH(J) + HRT(N4)
   32 DELH(J) = DELH(J)*TRAL
C
      LSP3 = LS + 3
C     RESET STANDARD OPTIONS
   33 MOLEF = .TRUE.
      MMHG = .FALSE.
      NEWPRT = .FALSE.
C
C     INITIALIZE
C     END    = 0.
      DELP = 0.
      NTB = 0
      NPRNTS = 0
      DO 34 I=1,50
   34 APRINT(I) = 0.
      AREA = 0.
      MDOT = 0.
C     P = 0.
      V = 0.
      RHO = 0.
C     T = 0.
      HN = 0.
      ERRN = 0.
      NH = 0
      AVH = 0.
      JCV = 30
      KOUNT = 0
      DO 35 I=1,28
      RK(I) = 0.
   35 F(I) = 0.
C     I1      SET IN -BLKRDA-
      I2 = 0
C
C     NAME OF INDEPENDENT VARIABLE - NAME OF ASSIGNED VARIABLE -
C     INPUT UNITS - OUTPUT UNITS - - SET IN NAMBLK
C
      IF (VERSA .EQ. BLANK) VERSA = AREAV
C
      IF (ACTION .NE. NEW) GO TO 80
C     INITIALIZE STEP SIZE LIMITS
      IF (VERSI .EQ. TIMEV) GO TO 78
      HMIN = 0.0001
      HMAX = 0.1000
      IPRCOD = 2
      GO TO 79
   78 HMIN = 0.500E-07
```

```
      HMAX = 0.500E-04
      IPRCOD = 4
   79 IF (VERSA .EQ. AREAV) IPRCOD = IPRCOD - 1
C
C    READ INTEGRATION CONTROLS, PROFILE OPTIONS,
C     PRINT OPTIONS, SPECIALTY SWITCHES
   80 IF( FIRSTC) READ (5,PROB)
C
      IF (.NOT. ALLM1) GO TO 36
      DO 77 I=1,25
      DO 77 J=1,30
   77 M(I,J) = 1.
      GO TO 40
C     THIRD BODY RATIOS SET IN -BLKRDA-
   36 CONTINUE
C
C    SET INITIAL STEP SIZE
      IF( FIRSTC ) GO TO 40
      HMIN  = HMINZ
      HMAX  = HMAXZ
      HINT  = HINTZ
      EMAX  = EMAXZ
      NPRINT = NPRNTZ
      EVSTEP= EVSTPZ
      DBUGO = DBUGOZ
   40 HI=HINT
      IF(HINT.EQ.0.)HI=HMIN
C
C    GET INITIAL CONDITIONS
C     CALL INIT(TSS,MMHG,MOLEF)
      IF( .NOT.FIRSTC ) GO TO 41
      HMINZ = HMIN
      HMAXZ = HMAX
      HINTZ = HINT
      EMAXZ = EMAX
      NPRNTZ= NPRINT
      EVSTPZ= EVSTEP
      DBUGOZ= DBUGO
      CALL MOVE(1,PRINT,PRINTZ,50,1)
   41 CONTINUE
      DO 39 J=1,LR
   39 B(J) = EACT(J)/1.987165
C
C     SPECIES TO BE NEGLECTED FROM ERROR CONSIDERATIONS SET IN -BLKRDA-
C     IT      SET IN -BLKRDA-
C
C    CHECK INPUT COMPOSITION
      CSUM = 0.
      DO 47 I=1,LS
   47 CSUM = CSUM + C(I)
```

```
      IF (ABS(1.-CSUM) .LE. .001) GO TO 48
      WRITE (6,105) CSUM,((SPNM(K,I),K=1,2),C(I),I=1,LS)
  105 FORMAT (7H0(KINP),5X,33HINVALID INPUT COMPOSITION  SUM = ,F11.6//
     * (12X,2A4,F20.5))
      NEXT = .TRUE.
      RETURN
C
   48 IF(ITPSZ.GT.2)GO TO 53
      IF (ITPSZ .EQ. 1  .AND.  NTB .EQ. 0) GO TO 53
      IF (NTB .NE. 0) NT = NTB
      CONV = 1.
      CON2 = 1.
      IF (VERSA .NE. AREAV) GO TO 203
      XU = CU
      AU(1) = CUA(1)
      AU(2) = CUA(2)
C     CONVERT AREA PROFILE TO INTERNAL UNITS
      IF (UNITI .NE. FPS) GO TO 201
      XU = FU
      AU(1) = FUA(1)
      AU(2) = FUA(2)
      CONV = 30.48
      GO TO 202
  201 IF (UNITI .NE. SI) GO TO 206
      XU = SU
      AU(1) = SUA(1)
      AU(2) = SUA(2)
      CONV = 100.
  202 CON2 = CONV*CONV
      GO TO 206
C
  203 XU = CU
      AU(1) = CUP2(1)
      AU(2) = CUP2(2)
C     CONVERT PRESSURE PROFILE TO INTERNAL UNITS
      IF (UNITI .NE. FPS) GO TO 204
      XU = FU
      AU(1) = FUP(1)
      AU(2) = FUP(2)
      CONV = 30.48
      CON2 = 1./2116.2
      GO TO 205
  204 IF (UNITI .NE. SI) GO TO 205
      XU = SU
      AU(1) = SUP(1)
      AU(2) = SUP(2)
      CONV = 100.
      CON2 = 1./1.01325E+05
  205 IF (.NOT. MMHG) GO TO 206
      AU(1) = CUP1(1)
```

```
      AU(2) = CUP1(2)
      CON2 = 1./760.
C
  206 IF (VERSI .EQ. TIMEV) CONV = 1.
      IF (ITPS2 .EQ. 2) GO TO 208
      DO 207 I=1,NTB
      CXTB(I) = XTB(I)*CONV
  207 CATB(I) = ATB(I)*CON2
      GO TO 53
  208 DO 209 I=1,4
  209 CN(I) = CX(I)*CON2
C
   53 IF ((NPRNTS .NE. 0) .OR. (DELP .NE. 0.) .OR. (END .NE. 0.)) NEWPRT
     * = .TRUE.
      IF (.NOT. NEWPRT) GO TO 59
      IF (END .NE. 0.) UNCEND = END
C     PREPARE PRINT STATIONS
      IF (EVSTEP) GO TO 59
      IF (NPRNTS .NE. 0) GO TO 57
      IF (DELP .NE. 0.) GO TO 54
      DELP = (UNCEND - IVAR)/24.9999
   54 PRINT(1) = IVAR + DELP
      DO 55 I=2,50
      PRINT(I) = PRINT(I-1) + DELP
      IF (PRINT(I) .GE. UNCEND) GO TO 56
   55 CONTINUE
   56 NP = I
      PRINT(NP) = UNCEND
      GO TO 59
C
   57 NP = NPRNTS
      IF (APRINT(1) .EQ. 0.) GO TO 59
      CONV = 1.
      IF (IPRCOD .EQ. 2 .OR. IPRCOD .EQ. 4) GO TO 210
      IF (UNITI .EQ. FPS) CONV = 30.48
      IF (UNITI .EQ. SI) CONV = 100.
      CON2 = CONV*CONV
      GO TO 213
  210 CON2 = 1.
      IF (UNITI .NE. FPS) GO TO 211
      CONV = 30.48
      CON2 = 1./2116.2
      GO TO 212
  211 IF (UNITI .NE. SI) GO TO 212
      CONV = 100.
      CON2 = 1./1.01325E+05
  212 IF (MMHG) CON2 = 1./760.
  213 IF (VERSI .EQ. TIMEV) CONV = 1.
      CALL CURS (CATB,CXTB,NT).
      DO 58 I=1,NPRNTS
```

```
      APRINT(I) = APRINT(I)*CON2                                        KINPP401
      CALL CINP (APRINT(I),PRINT(I),DUM1,DUM2)                          KINPP402
   58 PRINT(I) = PRINT(I)/CONV                                          KINPP403
C                                                                       KINPP404
   59 IF (TTPSZ .EQ. 1) CALL CUBS (CXTB,CATB,NT)                        KINPP405
C                                                                       KINPP406
      IF (UNITI .NE. FPS) GO TO 63                                      KINPP407
C    CONVERT FROM FPS UNITS TO INTERNAL (CGS) UNITS                     KINPP408
      IF (VERSI .NE. TIMEV) GO TO 60                                    KINPP409
      DVAR = DVAR*30.48                                                 KINPP410
      GO TO 61                                                          KINPP411
   60 IVAR = IVAR*30.48                                                 KINPP412
   61 IF (MMHG) P = P*2.7845                                            KINPP413
      P = P/2116.2                                                      KINPP414
      AREA = AREA*929.0304                                              KINPP415
      MDOT = MDOT*453.59237                                             KINPP416
      V = V*30.48                                                       KINPP417
      RHO = RHO/62.43                                                   KINPP418
      T = T/1.8                                                         KINPP419
      IF ((.NOT. NEWPRT)  .OR.  VERSI .EQ. TIMEV) GO TO 68              KINPP420
      CEND = UNCEND*30.48                                               KINPP421
      DO 62 I=1,NP                                                      KINPP422
   62 PRINT(I) = PRINT(I)*30.48                                         KINPP423
      GO TO 68                                                          KINPP424
   63 IF (UNITI .NE. SI) GO TO 67                                       KINPP425
C    CONVERT FROM SI UNITS TO INTERNAL (CGS) UNITS                      KINPP426
      IF (VERSI .NE. TIMEV) GO TO 64                                    KINPP427
      DVAR = DVAR*100.                                                  KINPP428
      GO TO 65                                                          KINPP429
   64 IVAR = IVAR*100.                                                  KINPP430
   65 IF (MMHG) P = P*133.3224                                          KINPP431
      P = P/1.01325E+05                                                 KINPP432
      AREA = AREA*10000.                                                KINPP433
      MDOT = MDOT*1000.                                                 KINPP434
      V = V*100.                                                        KINPP435
      RHO = RHO*.001                                                    KINPP436
      IF ((.NOT. NEWPRT)  .OR.  VERSI .EQ. TIMEV) GO TO 68              KINPP437
      CEND = UNCEND*100.                                                KINPP438
      DO 66 I=1,NP                                                      KINPP439
   66 PRINT(I) = PRINT(I)*100.                                          KINPP440
      GO TO 68                                                          KINPP441
C                                                                       KINPP442
   67 CEND = UNCEND                                                     KINPP443
      IF (MMHG) P = P/760.                                              KINPP444
C                                                                       KINPP445
   68 MIXMW = 0.                                                        KINPP446
C                                                                       KINPP447
      IF (.NOT. MOLEF) GO TO 71                                         KINPP448
C    MOLE FRACTION TO MOLES(I)/MASS(MIXTURE)                            KINPP449
      DO 69 I=1,IS                                                      KINPP450
```

```
   69 MIXMW = MIXMW + C(I)*MW(I)
      DO 70 I=1,IS
   70 SIGMA(I) = C(I)/MIXMW
      GO TO 73
C
C     MASS FRACTION TO MOLES(I)/MASS(MIXTURE)
   71 DO 72 I=1,IS
      SIGMA(I) = C(I)/MW(I)
   72 MIXMW = MIXMW + SIGMA(I)
      MIXMW = 1./MIXMW
C
C     UNIVERSAL GAS CONSTANT IN ATM-CC/MOLE-DEG K
   73 RR = 82.056
C     UNIVERSAL GAS CONSTANT IN ERGS/MOLE-DEG K
      R = 8.3143E+07
C
      IF (M2 .EQ. 0. .AND. .NOT. (COMBUS .OR. SHOCK)) GO TO 81
      CALL THRM (T,1.)
      CPRO = 0.
      DO 74 I=1,IS
   74 CPRO = CPRO + CPR(I)*SIGMA(I)
      GAMMA = CPRO/(CPRO - 1./MIXMW)
      IF (V .NE. 0.) GO TO 81
      V = SQRT(M2*R/MIXMW*GAMMA*T)
C
   81 IF (P .EQ. 0.) GO TO 82
      RHO = P*MIXMW/(RR*T)
      GO TO 75
   82 IF (RHO .EQ. 0.) GO TO 83
      P = RHO*RR*T/MIXMW
      GO TO 75
C
   83 IF (IPRCOD .GT. 2) GO TO 84
      X = TVAR
      IF (VERSI .EQ. TIMEV) X = DVAR
      CALL CINP (X,AVAR,DUM1,DUM2)
      GO TO 85
   84 TIME = DVAR
      IF (VERSI .EQ. TIMEV) TIME = TVAR
      CALL CINP (TIME,AVAR,DUM1,DUM2)
   85 IF (VERSA .EQ. AREAV) GO TO 86
      P = AVAR
      GO TO 81
   86 AREA = AVAR
      RHO = MDOT/(AREA*V)
      GO TO 82
C
   75 IF (MDOT .EQ. 0.) MDOT = RHO*AREA*V
C
C
```

```
      RETURN                                                                  KINPP501
      END                                                                     KINPP502
```

```
CLCFIT          INTEGRATE OR INTERPOLATE                         'LCFIT'
C
C     INTEGRATE OR INTERPOLATE USING A PARABOLA WH9CH PASSED THROUGH THE
C     AND (I+1) POINTS BUT MISSES THE (I-1) AND (I+2) POINTS (IF THEY BO
C     EXIST) SUCH THAT THE SQUARE OF THE DEVIATION IS A MINIMUM.  NOTE
C     THAT  I  IS GENERALLY SELECTED SUCH THAT
C               X(I).LE.XC.LT.X(I+1)
C     THE EQUATION FOR THE PARABOLA IS
C               Y-Y(I) = B*(X-X(I)) + C*(X-X(I))**2
C
      SUBROUTINE  LCFIT(X,Y,NPTS, NEW, XC,YC,NXC,ND, C)
       DIMENSION X(10),Y(10), XC(10),YC(10), C(10)
       LOGICAL NEW
C
C    INPUT-
C     X, Y      PTS. ON CURVE
C     NPTS      NO. OF X
C     NEW    =  .FALSE. IF THE 'C' ARRAY OF COEFFICIENTS IS
C               AVAILABLE FROM A PREVIOUS ENTRY
C     XC        LIST OF X AT WHICH CALC TO BE DONE
C     YC(1)     INTEGRATION CONSTANT IF  ND=-1
C     NXC       NO. OF XC
C     ND        =0 TO GET COORD,  =1 TO GET 1ST DERIVATIVE,
C               =-1 FOR INTEGRATION
C
C    OUTPUT
C     YC        COORDINATE OR DERIVATIVE AT XC    OR
C               YC(IC)= INTEGRAL(Y*DX) FROM XC(1) TO XC(IC) WHERE IC=2,NXC
C     C      =  ARRAY OF (NPTS-1) COEFFICIENTS
C
C    NOTES-
C     FOR INTEGRATION 'XC' MUST BE IN THE SAME ORDER AS 'X'.  FOR INTERP
C     'X' MAY BE IN EITHER ASCENDING OR DESCENDING ORDER.
C        NO SPECIAL ORDER IS REQUIRED
C
      COMMON /CLSPF / I
C
      LOGICAL WITHIN
      DATA BITS/O377777777777/
      IF(.NOT.NEW) GO TO 90
      N     = NPTS-1
C
C     EVALUATE COEFFICIENT  C(I)
C
      DO 30 I=1,N
      XI    = X(I)
      YI    = Y(I)
      X3    = X(I+1)-XI
      BOT   = 0.
      TOP   = 0.
```

```
      C(I)   = 0.                                                        LCFIT051
      Y3     = Y(I+1)-YI                                                 LCFIT052
      IF(I.LE. 1  .OR.  Y(I-1).EQ.BITS) GO TO 27                        LCFIT053
      X1     = X(I-1)-XI                                                 LCFIT054
      X13    = X(I-1)-X(I+1)                                             LCFIT055
      TOP    = X1*(Y3*X1-(Y(I-1)-YI)*X3)*X13                             LCFIT056
      BOT    = X1*X1*X13*X13*X3                                          LCFIT057
   27 IF(I.GE.N  .OR.  Y(I+2).EQ.BITS)  GO TO 28                        LCFIT058
  271 X4     = X(I+2)-XI                                                 LCFIT059
      X43    = X(I+2)-X(I+1)                                             LCFIT060
      TOP    = TOP + X4*(Y3*X4-(Y(I+2)-YI)*X3)*X43                       LCFIT061
      BOT    = BOT + X4*X4*X43*X43*X3                                    LCFIT062
   28 IF(BOT.NE.0.) C(I)=-TOP/BOT                                        LCFIT063
   30 CONTINUE                                                           LCFIT064
C                                                                        LCFIT065
C     BEGIN INTERPOLATION LOOP FOR XC(IC)  IC=1,NXC                      LCFIT066
   90 I      = MAX0(1,MINO(I,N))                                         LCFIT067
      IF(ND.EQ.(-1)) I=1                                                 LCFIT068
      SGN    = SIGN(1.,X(N+1)-X(1))                                      LCFIT069
      IC     = 1                                                         LCFIT070
      GO TO 160                                                          LCFIT071
C                                                                        LCFIT072
C     LOCATE APPROPRIATE INTERVAL                                        LCFIT073
  100 WITHIN= .FALSE.                                                    LCFIT074
      NCOUNT= N                                                          LCFIT075
  102 IF(NCOUNT) 119,103,103                                             LCFIT076
  103 NCOUNT= NCOUNT-1                                                   LCFIT077
C                                                                        LCFIT078
      XI     = X(I)                                                      LCFIT079
      XD     = XC(IC)-XI                                                 LCFIT080
      IF(N) 104,120,104                                                  LCFIT081
  104 IF(SGN*XD) 105,107,110                                             LCFIT082
C                                                                        LCFIT083
C       F.LT.0. (F IS THE FRACTIONAL POSITION IN THE INTERVAL)           LCFIT084
  105 IF(I.EQ.1)      GO TO 120                                          LCFIT085
      IF(ND.EQ.(-1)) GO TO 119                                           LCFIT086
      I      = I-1                                                       LCFIT087
      GO TO 102                                                          LCFIT088
C                                                                        LCFIT089
C       F.EQ.0                                                           LCFIT090
  107 IF(X(I+1).NE.XI) GO TO 120                                         LCFIT091
      GO TO 116                                                          LCFIT092
C                                                                        LCFIT093
C       F.GT.0.                                                          LCFIT094
  110 IF(SGN*(XC(IC)-X(I+1))) 120,112,114                                LCFIT095
C                                                                        LCFIT096
C       F.EQ.1.0, CHECK FOR INTEGRATION AND DOUBLE POINT BEFORE INCREMENLCFIT097
  112 IF((ND.EQ.(-1)) .OR. (I.NE.N .AND. X(I+1).EQ.X(I+2))) GO TO 120   LCFIT098
C       F.GT.1.0                                                         LCFIT099
  114 IF(I.EQ.N) GO TO 120                                               LCFIT100
```

```
      IF(ND.EQ.(-1)) GO TO 125
  116 I     = I+1
      GO TO 102
C
  119 CALL ERROR1
C
C    PRELIMINARY CALCULATIONS FOR INTERPOLATION OR INTEGRATION
  120 WITHIN=.TRUE.
  125 X3     = X(I+1)-X(I)
      YI     = Y(I)
      B      = 0.
      CI     = C(I)
      IF(N.GT.0 .AND. X3.NE.0.)  B = (Y(I+1)-YI)/X3 - CI*X3
  129 IF(ND) 130,140,141
C
C    ND=-1, INTEGRATE
  130 IF(.NOT.WITHIN) XD=X3
      S1     = (YI + (B/2. + CI/3.*XD)*XD)*XD
      IF(WITHIN) GO TO 135
C    'I' IS BEING INCREMENTED TO FIND APPROPROATE INTERVAL.  HENCE,
C      CUMULATE THE INTEGRAL OF THE ITH INTERVAL.
      SA     = SA + S1
      GO TO 116
C    APPROPRIATE INTERVAL FOUND.    X(I)-XC(IC)-X(I+1)
  135 IF(IC.EQ.1) SA=YC(IC)-S1
      IF(IC.NE.1) YC(IC)=SA+S1
      GO TO 150
C
C    ND=0,  INTERPOLATE FOR COORDINATES
  140 YC(IC)= YI + (B + CI*XD)*XD
      GO TO 150
C
C    ND=1,  FIRST DERIVATIVE
  141 YC(IC)= B + 2.*CI*XD
      GO TO 150
C
  150 IC     = IC+1
  160 IF(NXC-IC) 900,100,100
C
  900 RETURN
      END
```

```
CLESVV       GEN'L DBL PREC LINEAR EQN SOLVER
      SUBROUTINE LESV (K)
C
C     THIS ROUTINE IS A GENERAL DOUBLE PRECISION LINEAR EQUATION SOLVER
C       IN THIS PROGRAM IT IS USED TO COMPUTE THE INCREMENTS K(I)
C
      DOUBLE PRECISION A,S,TS,B
C
      LOGICAL NEXT
C
      REAL K
C
      DIMENSION S(28),K(28)
C
      COMMON/COND/DUMMY(34),N,NEXT
      COMMON/MATX/A(28,29)
C
      NP = N+1
      DO 5 I=1,N
C     GET SCALE FACTORS
      TS = 0.D0
      DO 6 J=1,N
      B = DABS(A(I,J))
      IF (B .GT. TS) TS = B
    6 CONTINUE
      IF (TS .EQ. 0.D0) GO TO 100
C
C     SCALE ROWS
      DO 10 J=1,NP
   10 A(I,J) = A(I,J)/TS
    5 CONTINUE
C
C     BEGIN TRIANGULARIZATION
      IF (N .EQ. 1) GO TO 25
      NM = N-1
      DO 15 J=1,NM
C     FIND MAXIMUM ELEMENT IN COLUMN J BELOW DIAGONAL
      II = J
      JP = J+1
      DO 16 I=JP,N
      IF (DABS(A(I,J)) .GT. DABS(A(II,J))) II = I
   16 CONTINUE
      IF (II .EQ. J) GO TO 20
C     INTERCHANGE ROWS II AND J
      DO 17 L=J,NP
      TS = A(II,L)
      A(II,L) = A(J,L)
   17 A(J,L) = TS
C
C     ZERO COLUMN J BELOW DIAGONAL
```

```
   20 DO 18 I=JP,N
      TS = A(I,J)/A(J,J)
      IF (TS .EQ. 0.D0) GO TO 18
      DO 19 L=JP,NP
   19 A(I,L) = A(I,L) - TS*A(J,L)
   18 CONTINUE
   15 CONTINUE
C
C     BACK SUBSTITUTE
   25 TS = A(N,N)
      IF (DABS(TS) .LT. 1.D-10) WRITE (6,102)
  102 FORMAT (7H0(LESV),5X,11HSINGULARITY)
      S(N) = A(N,NP)/TS
      K(N) = S(N)
      IF (N .EQ. 1) RETURN
      DO 26 I=2,N
      M = NP - I
      TS = A(M,M)
      IF (DABS(TS) .LT. 1.D-10) WRITE (6,102)
      B = A(M,NP)
      MP = M+1
      DO 27 L=MP,N
   27 B = B - A(M,L)*S(L)
      S(M) = B/TS
   26 K(M) = S(M)
   30 CONTINUE
      RETURN
C
  100 WRITE (6,101) I
  101 FORMAT (7H0(LESV),3X,3HROW,I4,39H OF THE COEFFICIENT MATRIX IS AL
     1 ZEROS)
      DO 50 I=1,N
   50 WRITE (6,103) (A(I,J), J=1,NP)
  103 FORMAT (1H1,8E16.6/(1X,8E16.6))
      NEXT = .TRUE.
      RETURN
C
      END
```

```
*LSPFIT       DUMMY--CALLS LFIT                                    LSPFIT01
      SUBROUTINE LSPFIT (X,Y,N,XC,YC,NP,I)                         LSPFIT02
      CALL LFIT (X,Y,N,XC,YC,NP,I)                                 LSPFIT03
      RETURN                                                       LSPFIT04
      END                                                          LSPFIT05
```

```
CLFIT
      SUBROUTINE LFIT(X,Y,NPTS,XC,YC,NXC,ND)
CLFIT         INTEGRATE OR INTERPOLATE
*     INTEGRATE OR INTERPOLATE USING A STRAIGHT LINE FIT BETWEEN POINTS
C
C
C*  CHECK TO KEEP -LFIT- FROM EXTRAPOLATION--
C*  IF REQUESTED INTERPOLATION LOCATION IS OUTSIDE DEFINED INTERVAL, SET
C*  VARIABLE EQUAL TO END VALUE
C
C
C
      DIMENSION X(10),Y(10), XC(10),YC(10)
C
C     INPUT-
C      X, Y      PTS. ON CURVE
C     NPTS       NO. OF X
C     XC         LIST OF X AT WHICH CALC TO BE DONE
C     YC(1)      INTEGRATION CONSTANT IF ND=-1
C     NXC        NO. OF XC
C     ND         =0 TO GET COORD,  =1 TO GET 1ST DERIVATIVE,
C                =-1 FOR INTEGRATION
C
C    OUTPUT
C     YC         COORDINATE OR DERIVATIVE AT XC    OR
C                YC(IC)= INTEGRAL(Y*DX) FROM XC(1) TO XC(IC) WHERE IC=2,NX
C
C    NOTES
C     'X' MAY BE IN EITHER ASCENDING OR DESCENDING ORDER.
C     FOR INTEGRATION 'XC' MUST BE IN THE SAME ORDER AS 'X'.
C     FOR INTERPOLATION NO SPECIAL ORDER IS REQUIRED.
C
      COMMON /CLSPF /  I
      LOGICAL  WITHIN
C
      N      = NPTS-1
      I      = MAX0(1,MIN0(I,N))
      IF (ND.EQ.(-1)) I=1
      ISAVE = 0
      SGN    = SIGN(1.,X(N+1)-X(1))
C
C    BEGIN INTERPOLATION LOOP FOR XC(IC)  IC=1,NXC
      IC     = 1
C
C  LOCATE APPROPRIATE INTERVAL
  100 WITHIN= .FALSE.
      NCOUNT= N
  102 IF (NCOUNT) 119,103,103
  103 NCOUNT= NCOUNT-1
      XI     = X(I)
```

```
      XD     = XC(IC)-XI
      IF (N) 104,120,104
  104 IF (SGN*XD)105,107,110
C
C     F.LT.0. (F IS THE FRACTIONAL POSITION IN THE INTERVAL)
C
C
  105 IF (I.EQ.1) GO TO 117
C*105 IF (I.EQ.1) GO TO 120
C
C
      IF (ND.EQ.(-1)) GO TO 119
      I      = I-1
      GO TO 102
C
C     F.EQ.0.
  107 IF (X(I+1).NE.XI) GO TO 120
      GO TO 116
C     F.GT.0.
  110 IF (SGN*(XC(IC)-X(I+1))) 120,112,114
C
C     F.EQ.1.,  CHECK FOR INTEGRATION AND DOUBLE POINT BEFORE INCREMENTIN
  112 IF ((ND.EQ.(-1)) .OR. (I.NE.N .AND. X(I+1).EQ.X(I+2))) GO TO 120
C
C     F.GT.1.
C
C
  114 IF (I.EQ.N) GO TO 118
C*114 IF (I.EQ.N) GO TO 120
C
C
      IF (ND.EQ.(-1)) GO TO 122
  116 I      = I+1
      GO TO 102
C
C
C
C*  EXTRAPOLATION OUTSIDE X-TABLE---USE END VALUES
  117 YI=Y(1)
      GO TO 1181
C
  118 YI=Y(N+1)
 1181 B=0.
C
C
      GO TO 129
C
C
C
  119 CALL ERROR1
```

```
*LFIT1          LINFAR FIT INTERPOLATION                      *LFI1*
      SUBROUTINE LFIT1(X,Y,NPTS, XC,YC,NXC)
       DIMENSION           X(10),Y(10), XC(10),YC(10)
C
C     INPUT-
C      X,Y    = LIST OF COORDINATES DESCRIBING THE INPUT FUNCTION
C      NPTS   = NUMBER OF X,Y POINTS
C      XC     = LIST OF X-S AT WHICH INTERPOLATION IS TO BE PERFORMED
C      NXC    = NUMBER OF XC-VALUES
C
C     OUTPUT-
C      YC     = LIST OF VALUES INTERPOLATED AT XC(IC),IC=1,NXC
C
C     NOTES-
C      IF  XC  IS OUTSIDE OF THE RANGE OF  X,  THE END VALUE OF  Y  IS SU
C         FOR YC.
C      X  MUST BE LISTED FROM SMALLEST TO LARGEST.
C      DOUBLE  X-POINTS ARE ALLOWED FOR A FUNCTION DISCONTINUITY.
C
      N       = NPTS
      I       = 1
C
C     BEGIN INTERPOLATION LOOP FOR  XC(IC),IC=1,NXC
      IC      = 1
   60 XCIC    = XC(IC)
      IF(N.GT.1) GO TO 100
      YC(IC)=Y(1)
      GO TO 190
C
  100 XG      = X(I+1)-XCIC
      IF(XG) 114,114,102
  102 XF      = XCIC-X(I)
      IF(XF) 110,120,120
C
C       F.LT.0. (F IS THE FRACTIONAL POSITION IN THE INTERVAL)
  110 I       = I-1
      IF(I) 100,111,100
  111 I       = 1
      YC(IC)= Y(1)
      GO TO 190
C
C       F.GE.1.
  114 I       = I+1
      IF(I-N) 100,115,100
  115 I       = N-1
      YC(IC)= Y(N)
      GO TO 190
C
C     INTERPOLATE
  120 YC(IC)= (Y(I)*XG+Y(I+1)*XF)/(XG+XF)
```

```
C                                                               LFIT1051
C      INDEX TO NEXT  XC(IC)                                   LFIT1052
  190 IC     = IC+1                                             LFIT1053
      IF(IC.LE.NXC) GO TO 60                                    LFIT1054
C                                                               LFIT1055
      RETURN                                                    LFIT1056
      END                                                       LFIT1057
```

```
C
C     PRELIMINARY CALCULATIONS FOR INTERPOLATION OR INTEGRATION
  120 WITHIN= .TRUE.
  122 IF (I-ISAVE) 124,129,124
  124 ISAVE = I
      YI    = Y(I)
      X3    = X(I+1)-XI
      B     = 0.
      IF (N.GT.0 .AND. X3.NE.0.) B=(Y(I+1)-YI)/X3
  129 IF (ND) 130,140,141
C
C     ND=-1, INTEGRATE
  130 IF (.NOT.WITHIN) XD=X3
      S1    = (YI+.5*B*XD)*XD
      IF (WITHIN) GO TO 135
C     'I' IS BEING INCREMENTED TO FIND APPROPRIATE INTERVAL.  HENCE,
C      CUMULATE THE INTEGRAL OF THE ITH INTERVAL.
      SA    = SA+S1
      GO TO 116
C     APPROPRIATE INTERVAL FOUND.  X(I)-XC(IC)-X(I+1)
  135 IF (IC.EQ.1) SA=YC(IC)-S1
      IF (IC.NE.1) YC(IC)=SA+S1
      GO TO 150
C     ND=0,  INTERPOLATE FOR COORDINATES
  140 YC(IC) = YI+B*XD
      GO TO 150
C     ND=1,  FIRST DERIVATIVE
  141 YC(IC) = B
C
  150 IC     = IC+1
      IF (NXC-IC) 900,160,160
  160 IF (ND.NE.(-1) .AND. XC(IC).EQ.XC(IC-1)) I=I+1
      GO TO 100
C
  900 RETURN
      END
```

```
CMAINBD         MAIN BLOCK DATA                                         MAINBD01
      BLOCK DATA                                                        MAINBD02
      COMMON /TROUBL/ DERR(4)                                           MAINBD03
      LOGICAL         DERR                                              MAINBD04
      COMMON /CBITS / BITS,BLANK                                        MAINBD05
C                                                                       MAINBD06
      DATA DERR/4*F/                                                    MAINBD07
      DATA BITS,BLANK/O377777777777,6H      /                           MAINBD08
      END                                                               MAINBD09
```

```
CMAINJ        MAIN JETMIX SUBROUTINE
      SUBROUTINE MAIN
      COMMON /ADAM01/ NAM(40)
      COMMON /TAG   / NAM1(50)
      COMMON /CNERR / BITS1,KERR,DUMV(3)
      LOGICAL         KERR
      COMMON /CBITS / BITS,BLANK
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
      LOGICAL         ERR,ERRMAJ,INERR,PRERR
      COMMON /CARRY / NEW
      LOGICAL         NEW
      COMMON /FILK  / CSC
      COMMON /DIFE01/ DIF(122)
      COMMON /BCMOL / BCM(24)
C
      BITS1 = BITS
      NEW   = .TRUE.
      CSC   = 1000.
      KERR  = .FALSE.
      CALL SETM(1,BLANK,NAM1(21),30)
      CALL MOVE(1,NAM,NAM1,20,1)
C
      CALL LLINK(6HINLINK)
      CALL JTINIT
C
C     READ NAMELIST $A
C
    1 CALL JETINP
      IF( .NOT.KERR ) GO TO 2
      ERR   = .TRUE.
      RETURN
C
    2 CALL LLINK(6HCPLINK)
      CALL JTCTRL
      IF( KERR ) ERR=.TRUE.
      GO TO 3
C
      ENTRY   ERROR1
      ERR   = .TRUE.
    3 CALL JTOUTS
C
      RETURN
      END
```

```
CMAINP       MAIN PROGRAM--EMIS DRIVER
      LOGICAL EOF
      COMMON /IOFILE/ TAPIN,TAPOT
      LOGICAL          TAPIN,TAPOT
      COMMON /CPROGM/ PROGM
      INTEGER          PROGM
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
      LOGICAL          ERR,ERRMAJ,INERR,PRERR
      COMMON /ADAM02/ ENDJOB,DUMMY(3)
      LOGICAL          ENDJOB
      COMMON /ADAM01/ NAME(10),ADDRES(10),TITLE(10),IDENT(10)
      COMMON /CNFILE/ NF
      INTEGER PREJET,GMOD,SPALDG,EMIS,CFILE
      DATA PREJET,JETMIX,GMOD,SPALDG,EMIS,CFILE/
     * 6HPREJET,6HJETMIX,4HGMOD,6HSPALDG,4HEMIS,5HCFILE/
C
C
      CALL FLGEOF(5,EOF)
C
C     READ ADAM01 BLOCK
C
    5 READ (5,20) NAME
      READ (5,20) ADDRES
      READ (5,20) IDENT
   20 FORMAT(1X,10A6)
   22 TAPOT = .FALSE.
      TAPIN = .FALSE.
      READ (5,25) PROGM,TAPIN,TAPOT
      IF( EOF ) GO TO 1000
   25 FORMAT(1X,A6,4X,L1,1X,L1)
   30 WRITE (6,35) PROGM,TAPIN,TAPOT
   35 FORMAT(1H1,10X,16HEXECUTING PROGM=,A6/
     * 10X,6HTAPIN=,L2,5X,6HTAPOT=,L2/)
C
C    SELECT PROGRAM
C
   40 IGO   = 1
      IF( PROGM.EQ.JETMIX ) IGO=2
      IF( PROGM.EQ.GMOD ) IGO=3
      IF( PROGM.EQ.SPALDG ) IGO=4
      IF( PROGM.EQ.EMIS ) IGO=5
      IF( PROGM.EQ.CFILE ) IGO=6
   50 GO TO (100,200,250,300,400,500) , IGO
C
C     PREJET*****
C
  100 IF( TAPOT ) REWIND 1
  101 CALL LLINK( PREJET )
      CALL MAIN
      IF( TAPOT ) ENDFILE 1
```

```
      IF( ERR ) GO TO 600                                               MAINP051
      GO TO 22                                                          MAINP052
C                                                                       MAINP053
C     JETMIX*****                                                       MAINP054
C                                                                       MAINP055
  200 IF((.NOT.TAPIN).AND.(.NOT.TAPOT)) GO TO 201                       MAINP056
      REWIND 1                                                          MAINP057
      REWIND 2                                                          MAINP058
      REWIND 3                                                          MAINP059
  201 CALL LLINK( JETMIX )                                              MAINP060
      CALL MAIN                                                         MAINP061
      IF( TAPOT ) ENDFILE 2                                             MAINP062
      IF( ERR ) GO TO 600                                               MAINP063
      GO TO 22                                                          MAINP064
C                                                                       MAINP065
C     GMOD*******                                                       MAINP066
C                                                                       MAINP067
  250 REWIND 1                                                          MAINP068
      REWIND 3                                                          MAINP069
      CALL LLINK(4HGMOD)                                                MAINP070
      CALL MAIN                                                         MAINP071
      ENDFILE 1                                                         MAINP072
      GO TO 22                                                          MAINP073
C                                                                       MAINP074
C     SPALDG*****                                                       MAINP075
C                                                                       MAINP076
  300 IF( TAPIN ) REWIND 2                                              MAINP077
      IF( TAPIN ) REWIND 1                                              MAINP078
      IF( TAPOT ) REWIND 3                                              MAINP079
  301 CALL LLINK( SPALDG )                                              MAINP080
      CALL MAIN                                                         MAINP081
      IF( TAPOT ) ENDFILE 3                                             MAINP082
      IF( ERR ) GO TO 600                                               MAINP083
      GO TO 22                                                          MAINP084
C                                                                       MAINP085
C     EMIS*****                                                         MAINP086
C                                                                       MAINP087
  400 REWIND 1                                                          MAINP088
      REWIND 2                                                          MAINP089
      REWIND 3                                                          MAINP090
  401 CALL LLINK( EMIS )                                                MAINP091
      CALL MAIN                                                         MAINP092
      IF( ERR ) GO TO 600                                               MAINP093
      GO TO 22                                                          MAINP094
C                                                                       MAINP095
C     CFILE*****                                                        MAINP096
C                                                                       MAINP097
  500 REWIND 1                                                          MAINP098
      REWIND 2                                                          MAINP099
      REWIND 3                                                          MAINP100
```

```
      IF( TAPOT ) REWIND 4
      IF( TAPIN ) REWIND 7
  501 CALL LLINK( CFILE )
      CALL MAIN
      IF(ERR.AND.PROGM.EQ.CFILE) GO TO 600
      REWIND 1
      REWIND 2
      REWIND 3
      IF(ERR.AND.(PROGM.EQ.JETMIX.OR.PROGM.EQ.SPALOG)) GO TO 1002
      GO TO 22
C
C
C
  600 WRITE (6,601) PROGM
  601 FORMAT(//2X,9HERR    = T,5X,6HPROGM=,A6//
     * 16X,21H** RUN TERMINATION **)
      IF(TAPOT.AND.(PROGM.EQ.JETMIX .OR. PROGM.EQ.SPALOG)) GO TO 500
 1000 WRITE(6,1001)
 1001 FORMAT(//////10X,26H********* ENDJOB *********)
 1002 STOP
      END
```

```
CMFMAIN     -MFMAIN- MAIN SUB. FOR MXFLUT
C
C
C     NOV. 1973
C
      SUBROUTINE MFMAIN
C
C**** COMMON /CINPUT/ XX(5),R(5,50),U(5,50),RHO(5,50),XSPECI(5,50,7),
      COMMON /CINPUT/ XY(5),R(5,50),U(5,50),RHO(5,50),XSPECI(5,50,13),
     1PPSI(5,50),GG(5,50),NJJ(5),NII
      COMMON /CSPECI/ NSPECI,NF,DX
      COMMON /JETDAT/ NTUBES,YREACT(12),TS(12),UREACT(12),SPV(12),
     1ZMWT(12),CP(12),FSPECI(12),GHAT(12),TKE(12),OTHER(36)
      COMMON /GASCMP/ GAS(1104)
      COMMON /CAXIAL/ X,XL,ALOGX
      COMMON /CXLOCA/ XX(5)
      COMMON /OPCTRL/ TITLE(20),PPRINT(30)
      COMMON /CINPJT/ DIAJ,DUMIX(114)
      COMMON /STCTRL/ LSTA,FINAL,CHEMK,FIRSTM,DUMST1(3),NIST,POUT1,
     *                ALFLIM,CMIXST,DUMST2(6)
      LOGICAL FIRSTM,FINAL
      COMMON /PORIDE/ PSTA(30)
      COMMON /CBITS/ BITS,BLANK
      COMMON /CDXSAV/ DXSAVE
      COMMON /XISAVE/ IXXX,ISTARX
      COMMON /CMXFLT/ XFRACT(12,12),XFRHAT(13,12),ALPHA(12,12)
      COMMON /CPRINT/ PDUM(20)
      DATA NSX/0/
C
      NAMELIST /INPUT/ X,XL,NTUBES,DX,ALFLIM,CMIXST,NIST
      IF (.NOT.FIRSTM) GO TO 250
      X=0.
      NQ=0
      XL=1.2
      DX=.05
      NIST  = 10
      READ (5,INPUT)
      DXSAVE=DX
      NII=5
      CMIXST= 2.
  100 CALL READT
      IF( PSTA(1).EQ.BITS ) GO TO 200
      DO 101 J=1,30
      IF( PSTA(J).EQ.BITS ) GO TO 200
      PPRINT(J)= PSTA(J)*DIAJ/12.
  101 CONTINUE
  200 ALOGX=ALOG(X+1.)
      CALL NEWNET
      IF( PDUM(19).NE.0. .AND. X.GE.PDUM(19) ) PDUM(17)=1.
      CALL MXFLUT
```

```
      IF( PDUM(18).NE.0. ) CALL MPRINT
      IF( FIRSTM ) GO TO 250
C
C     ADJUST STEP SIZE
C
      ALFLIM= .95
      NTUBE1= NTUBES-1
      ALFMIN= 2.
      DO 240 J=1,NTUBE1
      ALFMIN= AMIN1(ALFMIN,ALPHA(J,J) )
  240 CONTINUE
      IF( ALFMIN.GT.ALFLIM ) DXSAVE= CMIXST*DXSAVE
  250 XL= PPRINT(LSTA)*12./DIAJ
      FIRSTM= .FALSE.
      IF (X.GE.XL) GO TO 310
      NSX    = NSX+1
      IF( NSX.GT.NIST ) GO TO 311
      DX=DXSAVE
      X=X+DX
      IF (X.LE.XL) GO TO 260
      DX=XL-(X-DX)
      X=XL
  260 IF (X.LT.XX(NII-2)) GO TO 200
      IF(NII.LE.4 .AND. X.LT.XX(NII)) GO TO 200
C*  MOVE OLD DATA BACK IN STORAGE AND READ IN NEW DATA
      DO 300 I=1,4
      XX(I)=XX(I+1)
      XY(I)=XY(I+1)
      DO 299 J=1,50
      R(I,J)=R(I+1,J)
      U(I,J)=U(I+1,J)
      RHO(I,J)=RHO(I+1,J)
      DO 298 K=1,NF
  298 XSPECI(I,J,K)=XSPECI(I+1,J,K)
      GG(I,J)=GG(I+1,J)
  299 PPSI(I,J)=PPSI(I+1,J)
  300 NJJ(I)=NJJ(I+1)
      GO TO 100
C
C*  --FINAL=T-- INDICATED WE HAVE REACHED A PRINT STATION
  310 FINAL = .TRUE.
      RETURN
  311 XDIFF   = XL-X
      IF( XDIFF.LE.6.E-7 ) GO TO 310
      RETURN
      END
```

```
CMSHCUT         MESH REFINEMENT ROUTINE
      SUBROUTINE MSHCUT(NREG,YM,NPD)
      LOGICAL MCHANG
      COMMON /PROF/
     * DUM(400),UD(200),THD(200),ED(200)
      COMMON /UMESH/ MCHANG,CK,DY1,NMSH,           CXPC,CXTP,NRED
      COMMON /MISC/ PM(10)
C*********C*
C*
      COMMON /CTRL2/  DUM24(2), MERGE , DUM26(6)
      COMMON /BCO/ UO, EO, THO
      COMMON /JETTWO/ TWO, DUM55(7)
      LOGICAL TWO, MERGE
C*
      COMMON /MIXER/ MIX,RD(100),XD(100),CF,YR(100)
      LOGICAL  MIX
      COMMON /FLOBAL/ MAXIT,SUPB,NIT,PSID,YDD,YDC,
     *                P1,P2,UCL,TOL,UPSTRM,CVG
      LOGICAL  SUPB,CVG,UPSTRM
      COMMON /CNERR/ BITS,ERR,GC,GCJ,FOOT
      LOGICAL ERR
      COMMON /ACONVG/ YCD(100),PD(100),INDC(100),  CHOKE, CHOKED
      LOGICAL  CHOKE, CHOKED
      COMMON /DFIT/  CLSP(100)
      COMMON /STA2/ MACH2,TS2,SS2,V2,RHO2,DPDX2
      REAL  MACH2
      COMMON /BCMIX2/ GRADU,TW,MUW,RHOW,PTF,TTE
      REAL  MUW
      COMMON /CBODY/  YCB(100),CLSPCB(100),YCB1 , UCL1
C*
C*********C*
      DIMENSION IMP(10)
      EQUIVALENCE (IMP(1),PM(1))
      DIMENSION YM(1)
C*
C*  REDUCE NO. OF POINTS IN DOWNSTREAM MESH BY NRED
C*
      IF(IMP(1).EQ.0) GO TO 1
      CALL TABPRT(4HPSIB,YM,NPD,10,0)
      CALL TABPRT(2HUD,UD,NPD,10,0)
      CALL TABPRT(3HTHD,THD,NPD,10,0)
      CALL TABPRT(2HED,ED,NPD,10,0)
    1 KGO=NREG
   10 YE=YM(NPD)
      IF(.NOT.  MCHANG) GO TO 11
      GO TO (11,20,20), KGO
C*
C*  POTENTIAL CORE REGION--
C*  RECALCULATE NEW UNIFORM MESH.
C*
```

```
   11 LK=2                                                            MSHCUT51
      IF(.NOT. TWO)  GO TO 6312                                       MSHCUT52
      TESTL=U0-1.E-6                                                  MSHCUT53
      TESTU=U0+1.E-6                                                  MSHCUT54
      DO 6311 L=1,NPD                                                 MSHCUT55
      IF(UD(L).GE.TESTL  .AND.  UD(L).LE.TESTU)  UD(L)=U0             MSHCUT56
 6311 CONTINUE                                                        MSHCUT57
 6312 IF(.NOT. MCHANG) GO TO 166                                      MSHCUT58
      DO 14 L=2,NPD                                                   MSHCUT59
      IF(UD(L).NE. UD(1))  GO TO 16                                   MSHCUT60
   14 CONTINUE                                                        MSHCUT61
   16 LK=L                                                            MSHCUT62
C*                                                                    MSHCUT63
      IF((.NOT. TWO) .OR. MERGE) GO TO 166                            MSHCUT64
      DO 17 L=LK,NPD                                                  MSHCUT65
      IF(UD(L).EQ.U0) GO TO 18                                        MSHCUT66
   17 CONTINUE                                                        MSHCUT67
   18 LK=L                                                            MSHCUT68
      DO 19 L=LK,NPD                                                  MSHCUT69
      IF(UD(L).NE. U0)  GO TO 167                                     MSHCUT70
   19 CONTINUE                                                        MSHCUT71
  167 LK=L                                                            MSHCUT72
  166 LK1=LK-1                                                        MSHCUT73
      NPD=NPD-NRED                                                    MSHCUT74
      DY=(YE-YM(LK1))/FLOAT(NPD-LK1)                                  MSHCUT75
      DO 15 L=LK,NPD                                                  MSHCUT76
   15 YM(L)=YM(L-1)+DY                                                MSHCUT77
      GO TO 100                                                       MSHCUT78
C*                                                                    MSHCUT79
C*  TRANSITION/SIMILAR REGION-- CALCULATE NEW DY-1                    MSHCUT80
C*                                                                    MSHCUT81
   20 NPD=NPD-NRED                                                    MSHCUT82
      STEPS=FLOAT(NPD-1)                                              MSHCUT83
      DYC=YE/((CK**STEPS-1.)*(CK-1.)                                  MSHCUT84
      DO 30 L=2,NPD                                                   MSHCUT85
      XXP=FLOAT(L-1)                                                  MSHCUT86
   30 YM(L)=DYC*(CK**XXP-1.)/(CK-1.)                                  MSHCUT87
C*                                                                    MSHCUT88
  100 IF(IMP(1).EQ.0) GO TO 101                                       MSHCUT89
      CALL TABPRT(4HPSIA,YM,NPD,10,0)                                 MSHCUT90
  101 RETURN                                                          MSHCUT91
      END                                                             MSHCUT92
```

```
CMXFLTI       INITIATE HETEROGENEOUS GAS MODEL AT EACH PROFILE SURVEY
C             POINT IN ENGINE EXHAUST PLANE.
C
      SUBROUTINE MXFLTO(J)
      DIMENSION Z1(11),SMWT(16),ZL(16),ZR(11),          QV1(8),QV2(8)
      REAL N,KF,MACHR,MACHL,MWTJ
      INTEGER ILGO
      LOGICAL ALFUEL
      COMMON/JETDAT/NPTJ,RADJ(12),TSJ(12),UJ(12),SPVJ(12),MWTJ(12),
     'CPJ(12),FUELJ(12),SPLOGJ(12),TKEJ(12),OTHRJ(12,3)
      COMMON/GASCMP/YYK(12,2),FK(2,12,2),HK(2,12,2),ZK(16,2,12,2),
     'HCINCP(12,2),OTHRK(2,12,2,4)
      COMMON/COTREM/YTOL,YO,DYDX,CTRMAX/DOTREM/NDOT
      COMMON/INDATA/N,HF,WAR,T2,BETA,T25,FAR5,EINO2C,P0,
     'RCO(11),RCO2(11),RHC(11),RNOX(11),PT(11),PS(11),BLOC(11),FICOC(11)
      DATA SMWT/ 1.008,16.000, 2.016,32.000,17.008,18.016,28.010,44.010,
     '            28.016,39.944,30.008,14.008,33.008,46.008,17.032,00.000/
      DATA ZR,EIHC,EINOX,H2,SL,WTL1,H25,SR,WTR,CPFR,DH,HFV,CPFV/23*0.0/
      DATA AFR,SPVR/2*0.0/
      AVG(A,B)=YY*A+(1.0-YY)*B
      FGAM(G)=SQRT(G*(2.0/(G+1.0))**((G+1.0)/(G-1.0)))
C
C    START THE FLIGHT RECORDER
C
      NDOT=40
      REWIND 40
C
C    REDUCE GAS ANALYSIS DATA
C
      CALL CAROL(N,WAR,RHC(J),RCO(J),RCO2(J),RNOX(J),FBARS,EIHC,FICO,
     'EINOX)
      RH2CO=RH2(RCO(J))/RCO(J)
      SMWT(16)=120.1+10.08*N
      BLOC1=FAR5/FBARS/(1.0-FICO*0.1*SMWT(16)/28.01-EIHC)-1.0
      BLOC(J)=AMAX1(BLOC(J),BLOC1)
      FBARS=FBARS/(1.0+FBARS+WAR)
      FMAX=0.001496457*SMWT(16)
      FMAX=FMAX/(1.0+FMAX+WAR)
C
C    LOCAL WET AIR ENTHALPY (BLOC IS LOCAL BY-PASS RATIO)
C
      CALL EQGAST(0.0,WAR,N,T2,PS(J),.FALSE.,.FALSE.,Z1,H2,
     '     WTL1,SL,SPVL,AFL,CPFL)
      CALL EQGAST(0.0,WAR,N,T25,PS(J),.FALSE.,.FALSE.,Z1,H25,
     '     WTL1,SL,SPVL,AFL,CPFL)
      HAS=(1.0+BETA)*H2-BETA*H25
      HAIR=(HAS+BLOC(J)*H25)/(1.0+BLOC(J))
C
C    VITIATED AIR COMPOSITION
C
```

```
      F5=FAR5/(1.0+BLOC(J))                                             MXFLTI51
      CALL SETM(1,0.0,Z1,11)                                            MXFLTI52
      Z1(11)=F5*FINO2C/46.008                                           MXFLTI53
      Z1(7)=F5*EICOC(J)/28.01                                           MXFLTI54
      Z1(3)=RH2CO*Z1(7)                                                 MXFLTI55
      Z1(10)=0.00032189                                                 MXFLTI56
      Z1(9)=0.02695794-0.5*Z1(11)                                       MXFLTI57
      Z1(6)=5.0AN*F5/SMWT(16)+WAR/18.016-Z1(3)                          MXFLTI58
      Z1(8)=10.0*F5/SMWT(16)+1.036E-5-Z1(7)                             MXFLTI59
      Z1(4)=0.00724264+WAR/36.032-Z1(8)-0.5*(Z1(6)+Z1(7)+Z1(11))        MXFLTI60
      CALL FMPYC(1,1.0/(1.0+F5+WAR),Z1,Z1,11)                           MXFLTI61
      WTAIR=0.0                                                         MXFLTI62
      DO 10 IS=1,11                                                     MXFLTI63
   10 WTAIR=WTAIR+Z1(IS)                                                MXFLTI64
      WTAIR=1.0/WTAIR                                                   MXFLTI65
C                                                                       MXFLTI66
C   MAXIMUM CO THAT MODEL CAN CREATE  .                                 MXFLTI67
C                                                                       MXFLTI68
      FG5=F5/(1.0+F5+WAR)                                               MXFLTI69
      A=(FBARS-FG5)/(1.0-FG5)                                           MXFLTI70
      EICOMX=(A/FBARS-EIHC)*10.0/SMWT(16)*28.01                         MXFLTI71
     '      +EICOC(J)*FG5*(1.0-A)/FBARS                                 MXFLTI72
      IF(EICO.LE.EICOMX)GO TO 30                                        MXFLTI73
      WRITE(6,9996)J,EICO,EICOMX                                        MXFLTI74
 9996 FORMAT(15H0MXFLTO - PT NO,I3,27H, CANNOT MODEL MEASURED CO ,      MXFLTI75
     '43HWITHOUT DISSOCIATING TURBINE DISCHARGE GAS,/10X,               MXFLTI76
     '17HEICO REDUCED FROM,1PE10.3,3H TO,E10.3)                         MXFLTI77
      EICO=EICOMX                                                       MXFLTI78
   30 CONTINUE                                                          MXFLTI79
C                                                                       MXFLTI80
C   MAXIMUM MASS FRACTION RICH GAS IN SAMPLE                            MXFLTI81
C                                                                       MXFLTI82
      YYMAX=1.0-FBARS*EIHC                                              MXFLTI83
C                                                                       MXFLTI84
C   YY AT WHICH RICH GAS IS STOICHIOMETRIC                              MXFLTI85
C                                                                       MXFLTI86
      FARS=0.1*SMWT(16)/28.966643*0.209495/(1.0+0.25*N)                 MXFLTI87
      FSTOIC=FARS/(1.0+FARS+WAR)                                        MXFLTI88
      YYSTC=(FBARS-FG5-(1.0-FG5)*FBARS*EIHC)/(FSTOIC-FG5)               MXFLTI89
      YYSTEP=0.1                                                        MXFLTI90
      IF(FBARS.LT.FSTOIC)YYSTEP=YYMAX-YYSTC                             MXFLTI91
C                                                                       MXFLTI92
C   INITIALIZE BEFORE ITERATING                                         MXFLTI93
C                                                                       MXFLTI94
      YYS=YYMAX                                                         MXFLTI95
      U=0.0                                                             MXFLTI96
      TL=1800.0                                                         MXFLTI97
      TR=3000.0                                                         MXFLTI98
      QV2(1)=0.0                                                        MXFLTI99
C                                                                       MXFLTI100
```

```
C     COMMENCE ITERATION FOR RICH FRACTION
C
   40 IF(YYS.GT.YYMAX)YYS=YYMAX
      A=FBARS*FIHC
      YYMIN=(FBARS-FG5-A*(1.0-FG5))/(FMAX-FG5)
      IF(YYS.LT.YYMIN)YYS=YYMIN
      FUNLIT=1.0
      IF(YYS.LT.YYMAX)FUNLIT=A/(1.0-YYS)
      FL=FUNLIT+FG5*(1.0-FUNLIT)
      FR=(FBARS-(1.0-YYS)*FL)/YYS
C
      HTL=(1.0-FL)*HAIR+FL*HF
      HTR=(1.0-FR)*HAIR+FR*HF
      FARR=FR*(1.0+WAR)/(1.0-FR)
      IF(FARR.LT.0.0)CALL ERROR1
      CALL EQGASH(FARR,WAR,N,TTR,PT(J),.FALSE.,.FALSE.,ZR,
     ' HTR,WTR,SR,SPVR,AFR,CPFR)
C
C     LEAN GAS COMPOSITION
C
      CALL SETM(1,0.0,ZL,16)
      CALL EMPYC(1,1.0-FUNLIT,Z1,ZL,11)
      ZL(16)=FUNLIT/SMWT(16)
      WTL=0.0
      DO 60 IS=1,16
   60 WTL=WTL+ZL(IS)
      WTL=1.0/WTL
      HL=HTL
      ASSIGN 50 TO TLGO
      GO TO 80
   50 TTRQL=TTR/TL
C
C     COMMENCE ITERATION FOR VELOCITY
C
      QV1(1)=0.0
   70 KE=U**2/50072.884
C
C     RICH GAS PROPERTIES
C
      HR=HTR-TTRQL*KE/(1.0+(TTRQL-1.0)*YYS)
      ZR(11)=(FBARS*FINOX/46.008-(1.0-YYS)*ZL(11))/YYS
      IF((ZR(11).LT.0.0).OR.(YYS.EQ.YYMIN)) ZR(11)=0.0
      ZR(9)=0.02695794*(1.0-FR)/(1.0+WAR)-0.5*ZR(11)
      CALL EQGASH(FARR,WAR,N,TR,PS(J),.FALSE.,.TRUE.,ZR,
     '    HR,WTR,SR,SPVR,AFR,CPFR)
      C=CPFR*WTR/1.98596
      GAMR=C/(C-1.0)
      MACHR=U/AFR
      TTR=TR+KE/CPFR
      PTRPS=PRAT(MACHR,GAMR)
```

```
C                                                                      MXFLTI151
C   LEAN GAS PROPERTIES                                                MXFLTI152
C                                                                      MXFLTI153
      HL=HTL-KE/(1.0+(TTROL-1.0)*YYS)                                  MXFLTI154
      ASSIGN 115 TO TLGO                                               MXFLTI155
   80 ALFUEL=FUNLIT.EQ.1.0                                             MXFLTI156
      DO 100 ITCNT=1,30                                                MXFLTI157
      IF(TL.LT.189.0)TL=189.0                                          MXFLTI158
      CALL FRGAST(TL,PS(J),Z1,HLX,WTL1,SI,SPVL,AFL,CPFL)               MXFLTI159
      CALL FUFL(N,TL,HFV,CPFV)                                         MXFLTI160
      A=1.0-FUNLIT                                                     MXFLTI161
      HLX=FUNLIT*HFV+A*HLX                                             MXFLTI162
      CPFL=FUNLIT*CPFV+A*CPFL                                          MXFLTI163
      SPVL=ZL(16)*10.7319)*TL/PS(J)+A*SPVL                             MXFLTI164
      DT=(HL-HLX)/CPFL                                                 MXFLTI165
      IF(ABS(DT).LT.0.05)GO TO 110                                     MXFLTI166
      IF((TL.EQ.189.0).AND.(DT.LT.0.0))GO TO 105                       MXFLTI167
      IF(ABS(DT).GT.500.0)DT=SIGN(500.0,DT)                            MXFLTI168
  100 TL=TL+DT                                                         MXFLTI169
  105 IF(ALFUEL)GO TO 110                                              MXFLTI170
      WRITE(6,9999)J,DT,FL,TL,HL,YYS,U                                 MXFLTI171
 9999 FORMAT(51H0MXFLTO - LEAN GAS TEMP ITERATION DID NOT CONVERGE./   MXFLTI172
     16H0PT NO.,I3,6H   DT=,1PE12.4,5H  FL=,E12.4,5H  TL=,E12.4,        MXFLTI173
     15H  HL=,E12.4,6H  YYS=,E12.4,4H  U=,E12.4)                       MXFLTI174
  110 GO TO TLGO,(50,115)                                              MXFLTI175
  115 C=CPFL*WTL/1.98596                                               MXFLTI176
      GAML=C/(C-1.0)                                                   MXFLTI177
      AFL=SQRT(49721.372*GAML/WTL*TL)                                  MXFLTI178
      MACHL=U/AFL                                                      MXFLTI179
      TTL=TL+KE/CPFL                                                   MXFLTI180
      PTLPS=PRAT(MACHL,GAML)                                           MXFLTI181
C                                                                      MXFLTI182
C   FREE STREAM MASS AND VOLUME FRACTIONS OF RICH GAS                  MXFLTI183
C                                                                      MXFLTI184
      A=FGAM(GAMR)/FGAM(GAML)*PTRPS/PTLPS*SQRT(WTR/WTL*TTL/TTR)        MXFLTI185
      B=SPVR/SPVL                                                      MXFLTI186
      YY=1.0+A*B*(1.0-YYS)/YYS                                         MXFLTI187
      YY=1.0/YY                                                        MXFLTI188
      TAU=B*YY/(1.0+YY*(B-1.0))                                        MXFLTI189
C                                                                      MXFLTI190
C   CALCULATED APPARENT IMPACT PRESSURE                                MXFLTI191
C                                                                      MXFLTI192
      PTPSX=(PTRPS*TAU**2+PTLPS*(1.0-TAU)**2)/(TAU**2+(1.0-TAU)**2)    MXFLTI193
      WRITE(40,9998)J,YYS,EICOX,U,PTPSX,EICOC(J),FL,HTL,TL,            MXFLTI194
     1MACHL,PTLPS,ZL,FR,HTR,TR,MACHR,PTRPS,ZR                          MXFLTI195
 9998 FORMAT(I32X/10H ****** J=,I3,6H  YYS=,F7.4,8H  EICOX=,1PE10.3,   MXFLTI196
     14H  U=,0PF8.1,8H  PTPSX=,1PE10.3,                                 MXFLTI197
     16H  QCO=,E10.3,8H  ******,34X/20X,3HFL=,                          MXFLTI198
     1E10.3,6H  HTL=,0PF8.1,5H  TL=,F8.1,8H  MACHL=,1PE10.3,8H  PTLPS=, MXFLTI199
     1E10.3,7H    XL=,29X/2X,13E10.3,/2X,3E10.3,100X/20X,3HFR=,E10.3,   MXFLTI100
```

```
     '6H  HTR=,0PF8.1,5H  TR=,F8.1,8H  MACHR=,1PF10.3,8H  PTRPS=,E10.3,
     '7H    XR=,29X/2X,11E10.3,20X)
      Y0=PT(J)/PS(J)
      YTOL=1E-4*Y0
      DYDX=1.0/AVG(AFR,AFL)
      CALL QIREM(U,PTPSX,-2000.,QV1)
      IF(QV1(1).NE.0.0)GO TO 70
C
C     CHECK THE CALCULATED SAMPLE CO CONTENT
C
      EICOX=(YYS*ZR(7)+(1.0-YYS)*ZL(7))*28.01/FBARS
      IF((YYS.EQ.YYMAX).AND.(EICOX.GT.EICO))GO TO 120
      Y0=ALOG(EICO)
      Y=-50.0
      IF(EICOX.GT.0.0)Y=ALOG(EICOX)
      YTOL=1E-3
      DYDX=0.0
      CALL QIREM(YYS,Y,YYSTEP,QV2)
      IF(QV2(1).NE.0.0)GO TO 40
C
C     AVERAGE POINT PROPERTIES FOR JETMIX
C
  120 FUELJ(J)=AVG(FR,FL)
      SPLOGJ(J)=AVG((FR-FUELJ(J))**2,(FL-FUELJ(J))**2)
      CPJ(J)=AVG(CPFR,CPFL)
      MWTJ(J)=1.0/AVG(1.0/WTR,1.0/WTL)
      SPVJ(J)=AVG(SPVR,SPVL)
C     144/1.98596/778.16=0.09318008
      TSJ(J)=0.09318008*PS(J)*SPVJ(J)*MWTJ(J)
      UJ(J)=U
C
C     DETAILED 2-PART HETEROGENEOUS GAS PROPERTIES FOR MXFLUT
C
      YYK(J,1)=YY
      FK(1,J,1)=FR
      FK(2,J,1)=FL
      HK(1,J,1)=HR
      HK(2,J,1)=HL
      HCINCP(J,1)=0.0
      CALL MOVE(2,ZL,ZK(1,2,J,1),16,1,ZR,ZK(1,1,J,1),11,1)
      CALL SETM(1,0.0,ZK(12,1,J,1),5)
C
      RETURN
      END
```

```
*MXFLUT
      SUBROUTINE MXFLUT
C
      LOGICAL METRO
      COMMON /CMATRX/ INTR1(12),INTR2    ,DET,IFACTR
      COMMON /CRFACT/ RHOREA(12)
      COMMON /CMXFLT/ XFRACT(12,12),XFRHAT(13,12),ALPHA(12,12)
      COMMON /CSPECI/ NSPECI,NF,DX
      COMMON /CYOLD/ YRICHO(12)
      COMMON /CNWNET/ FDUM(100),FUELFL(12),FDUMM(100)
      COMMON /CCHECK/ SPECIF(12),SPECIS(13),RATIO(12)
C
      COMMON /JETDAT/ NTUBES,YREACT(12),TS(12),UREACT(12),SPV(12),
     *ZMWT(12),CP(12),FGSPEC(12),GHAT(12),TKE(12),TFR(12),TFL,
     * TIM1(12),TIM2,OTHER1(10)
      COMMON /GASCMP/ YRICH(12),YPRIME(12),FOAIR(2,12,2),ENTH(2,12,2),
     * CONC(16,2,12,2),HCINCP(12,2), U(12,2),DUMSG2(168)
      COMMON /GASTMW/ DUMTM1(96),TAU(12,2),DUMTM2(48)
      COMMON /CMASS/ ZMASS(12),ZMASSM(12),FUELL(12),FUELR(12)
      COMMON /CPRINT/ PDUM(20)
      COMMON /STCTRL/ ISTA,DUMST(16)
      COMMON /CAXIAL/ XCC,DUMAX(2)
      COMMON /CAGAIN/ AGAIN,IENTRY
      DIMENSION ERROR(12)
      DIMENSION CONC1(768)
      EQUIVALENCE (CONC1(1),CONC(1,1,1,1))
C
      DATA NCONC /16/
      DATA JDIM,IOLD,LNEW,JRICH,JLEAN/12,1,2,1,2/
C
      DO 11 IC=1,768
      IF(CONC1(IC).LT.0. ) CONC1(IC)=0.
   11 CONTINUE
      NRFACT=NTUBES
      IENTRY=IENTRY+1
C
      IF (IENTRY.GT.1) GO TO 100
      CALL MOVE(1,UREACT,U(1,LOLD),12,1)
  100 CONTINUE
C
C*  --FGSPEC-- IS THE FUEL/GAS RATIO OF SPECIE *K*
C
C*  ADD ENTRAINED MASS FLOW TO TOTAL MASS OF AMBIENT GAS SPECIES
      SPECIS(NSPECI)=SPECIS(NSPECI )+XFRACT(NSPECI,NSPECI)
C
C*  SCALE FLOW OF EACH SPECI TO INSURE EXACT CONSERVATION
      DO 210 K=1,NSPECI
      SPECIF(K)=0.
C*  SUM FLOW OF SPECIES *K* OVER ALL TUBES
      DO 200 J=1,NTUBES
```

```
  200 SPECIF(K)=SPECIF(K)+XFRHAT(K,J)
C
  210 CONTINUE
      IF( PDUM(16).NE.0. ) CALL TABPRT(6HSPECIF,SPFCIF,NF,10)
C*   SAVE OIIGINAl AMOUNT OF FLOW OF SEECIES(K) IN **SPECTS(K)
C
      IF( PDUM(17).EQ.0. ) GO TO 2100
      WRITE (6,2)
      WRITE (6,6)
      WRITE (6,4)
      WRITE (6,5) (K,K=1,12)
      WRITE (6,1) (J,(XFRACT(K,J),K=1,12     ),J=1,NSPECI)
 2100 CONTINUE
C
      CALL ALFA2(NTUBES)
C
      IF( PDUM(17).EQ.0. ) GO TO 2200
      WRITE (6,2)
      WRITE (6,8)
      WRITE (6,9) (K,K=1,12)
      WRITE (6,3) (J,(ALPHA(K,J),K=1,12),J=1,NSPECI)
C
      WRITE (6,2)
      WRITE (6,7)
      WRITE (6,4)
      WRITE (6,5) (K,K=1,12)
      WRITE (6,1) (J,(XFRHAT(K,J),K=1,12     ),J=1,NSPECI)
 2200 CONTINUE
C
C
C*   **ZMASS** IS MASS FLOW IN (UBE -J-
C*   **FUELL** IS MASS OF FUEL IN TUBE -J- IN LEAN CLASS
C*   **FUELR** IS MASS OF FUEL IN TUBE -J- IN RICH CLASS
C*   **FLEAN** IS MIXTURE RATIO IN LEAN CLASS
C*   **FRICH** IS MIXTURE RATIO IN RICH CLASS
C
C
C
C*   --FOAIR-- IS FUEL TO GAS RATIO
C
C*   SET MASS FLOW IN DUMMY OUTER TUBE
      ZMASS(NSPECI)=XFRACT(NSPECI,NSPECI)
C*    RECONSTRUCT MASS FLOW IN EACH TUBE AT NEW STATION
      DO 220 J=1,NTUBES
      ZMASSH(J)= 0.
      DO 220 K=1,NSPECI
  220 ZMASSH(J)= ZMASSH(J)+ZMASS(K)*ALPHA(K,J)
      CALL MOVE (1,UREACT,U(1,LNEW),12,1)
      DO 411 J=1,NTUBES
      FUFLI=0.
```

```
      FUELR=0.
      ZMASSL=0.
      ZMASSR=0.
C
      DO 301 L=1,NSPECI
C
C*  PERFORM UST MIXING STEP
C*  CALCULATE MASS OF FUEL IN EACH TUBE AND EACH CLASS
      FUELL    =ZMASS(L)*ALPHA(L,J)*FQAIR(JLEAN,L,LOLD)*(1.-YRICH(L))+
     1FUELL
      FUELR    =ZMASS(L)*ALPHA(L,J)*FQAIR(JRICH,L,LOLD)*YRICH(L)+FUELR
C
C*  CALAULATE MASS MFLOW IN EACH TUBE AND EACH CLASS
      ZMASSL=ZMASSL+ZMASS(L)*ALPHA(L,J)*(1.-YRICH(L))
      ZMASSR=ZMASSR+ZMASS(L)*ALPHA(L,J)*YRICH(L)
  301 CONTINUE
C
C*  CHECK FOR FRROR IN MASS FLOW CALCULATION
      FRROR(J)=ZMASSR+ZMASSL-ZMASSH(J)
      XFRHAT(NF,J)=(FUELL+FUELR)/(ZMASSL+ZMASSR)
C
C*  CALAULATE INTERIM MIXTURE RATION
      HETRO=ZMASSL.GT.0.0
      FLPRIM = 0.0
      IF (  HETRO       )  FLPRIM = FUELL / ZMASSL
      FRPRIM    =FUELR    /ZMASSR
C
C*  CALCULATE INTERIM MASS FRACTION OF RICH SPECIES
      YINTRM=ZMASSR/(ZMASSR+ZMASSL)
C
C*  CALCULATE FIMAL MIXTURE RATIOS AT NEW X-STATION
      FQAIR(JLEAN,J,LNEW)=FLPRIM
      SMALLF=FQAIR(JLEAN,J,LNEW)-XFRHAT(NF,J)
      GGG    = AMIN1(GHAT(J), YINTRM*(FRPRIM-XFRHAT(NF,J))**2+
     *         (1.-YINTRM)*SMALLF**2)
      FQAIR(JRICH,J,LNEW)= XFRHAT(NF,J)-GGG/SMALLF
C
C*  CALCULATE FRACTION OF RICH SPECIES AT NEW X-STATION
      YPRIME(J)=-SMALLF/(FQAIR(JRICH,J,LNEW)-FQAIR(JLEAN,J,LNEW))
C
C*  ALPHA(J,L) IS THE FRACTION OF MASS IN TUBE 'J' AT STATION-X, WHICH E
C*  IN TUBE 'L' AT STATION X+DX
C
C*  UPDATE CONCENTRATIONS AT NEW X-STATION
C
      K=J
C
C*  **YYYY** IS PERCENT OF MASS FLOW IN LEAN TUBE THAT IS POURED INTO RI
      YYYY=0.0
      IF(YINTRM .NE. 1.0) YYYY=(YPRIME(K)-YINTRM)/(1.0-YINTRM)
```

```
      IF(YYYY.LT.0.0)  YYYY=0.0                                         MXFLUT51
C                                                                       MXFLUT52
      DO 400 I=1,NCONC                                                  MXFLUT53
      CONCRH=0.                                                         MXFLUT54
      CONCLN=0.                                                         MXFLUT55
      DO 397 KDUM=1,NSPECI                                              MXFLUT56
C                                                                       MXFLUT57
C*  THIS EQUATION ASSUMES CONC=LBM I/LBM MIX--NOT LBM I/MOLE MIX        MXFLUT58
      CONCLN=CONCLN+         CONC(I,JLEAN,KDUM,LOLD)*ALPHA(KDUM,K)*(1.- MXFLUT59
     1YRICH(KDUM))*ZMASS(KDUM)                                          MXFLUT60
      CONCRH=CONCRH+         CONC(I,JRICH,KDUM,LOLD)*ALPHA(KDUM,K)*YRICH( MXFLUT61
     1KDUM)*ZMASS(KDUM)                                                 MXFLUT62
      IF( I.EQ.16 ) TER(KDUM)= CONC(I,JRICH,KDUM,LOLD)*ALPHA(KDUM,K)*   MXFLUT63
     * YRICH(KDUM)*ZMASS(KDUM)                                          MXFLUT64
  397 CONTINUE                                                          MXFLUT65
      CONCRH=CONCRH+CONCLN*YYYY                                         MXFLUT66
      IF( I.EQ.16) TEL=CONCLN*YYYY                                      MXFLUT67
      CONC(I,JRICH,K,LNEW)=CONCRH/(YPRIME(K)*(ZMASSL+ZMASSR))           MXFLUT68
      CONC(I,JLEAN,K,LNEW)=CONC(I,JLEAN,K,LOLD)                         MXFLUT69
      IF(HETRO)CONC(I,JLEAN,K,LNEW)=CONCLN/ZMASSL                       MXFLUT70
  400 CONTINUE                                                          MXFLUT71
C                                                                       MXFLUT72
      ENTHLN=0.                                                         MXFLUT73
      ENTHRH=0.                                                         MXFLUT74
      DO 410 KDUM=1,NSPECI                                              MXFLUT75
      ENTHLN=ENTHLN+(ENTH(JLEAN,KDUM,LOLD)+U(KDUM,LOLD)**2/50075.614)   MXFLUT76
     1*ALPHA(KDUM,K)*(1.0-YRICH(KDUM))*ZMASS(KDUM)                      MXFLUT77
  410 ENTHRH=ENTHRH+(ENTH(JRICH,KDUM,LOLD)+U(KDUM,LOLD)**2/50075.614)   MXFLUT78
     1*ALPHA(KDUM,K)*YRICH(KDUM)*ZMASS(KDUM)                            MXFLUT79
      ENTHRH=ENTHRH+ENTHLN*YYYY                                         MXFLUT80
      ENTH(JRICH,K,LNEW)=ENTHRH/(YPRIME(K)*(ZMASSL+ZMASSR))             MXFLUT81
     1-U(K,LNEW)**2/50075.614                                           MXFLUT82
      ENTH(JLEAN,K,LNEW)=ENTH(JLEAN,K,LOLD)                             MXFLUT83
      IF(HETRO)ENTH(JLEAN,K,LNEW)=ENTHLN/ZMASSL                         MXFLUT84
     1-U(K,LNEW)**2/50075.614                                           MXFLUT85
      TIM2  = TAU(K,2)                                                  MXFLUT86
      CALL HYCARB(K)                                                    MXFLUT87
      TIM1(K)= TAU(K,2)                                                 MXFLUT88
C                                                                       MXFLUT89
C     CALCULATE H/C INCIPIENCIES                                        MXFLUT90
  411 CONTINUE                                                          MXFLUT91
      XC3= 3.*DX                                                        MXFLUT92
      IF(XCC.LE.XC3.AND.PDUM(12).NE.0.) CALL TABPRT(5HZMASS,ZMASS,12,10)MXFLUT93
      IF(XCC.LE.XC3.AND.PDUM(12).NE.0.) CALL TABPRT(5HYRICH,YRICH,12,10)MXFLUT94
      IF(XCC.LE.XC3.AND.PDUM(12).NE.0.)CALL TABPRT(6HYPRIME,YPRIME,12,10MXFLUT95
     * )                                                                MXFLUT96
      IF(XCC.LE.XC3.AND.PDUM(12).NE.0.)CALL TABPRT(4HGHAT,GHAT,12,10)   MXFLUT97
      IF(XCC.LE.XC3.AND.PDUM(12).NE.0.)CALL TABPRT(5HFQATR,FQATR,24,10) MXFLUT98
      IF(XCC.LE.XC3.AND.PDUM(12).NE.0.)CALL TABPRT(5HALPHA,ALPHA,144,10)MXFLUT99
      IF( PDUM(14).NE.0. .AND. XCC.LE.PDUM(14))                         MXFLUT00
```

```
      * CALL TABPRT(6HGASCMP,ENTH,864,10)                               MXFLUT01
      IF( PDUM(16).NE.0. ) WRITE (6,10) (XFRHAT(NF,J),J=1,NSPECI)       MXFLUT02
C                                                                       MXFLUT03
      DO 420 K=1,NTUBES                                                 MXFLUT04
  420 YRICH(K)=YPRIMF(K)                                                MXFLUT05
C*  FQAIR IS NOT RESET BETWEEN X-STEPS                                  MXFLUT06
      DO 299 L=1,NTUBES                                                 MXFLUT07
      FQAIR(JRICH,L,LOLD)=FQAIR(JRICH,L,LNEW)                           MXFLUT08
      FQAIR(JLEAN,L,LOLD)=FQAIR(JLEAN,L,LNEW)                           MXFLUT09
C                                                                       MXFLUT10
C                                                                       MXFLUT11
      U(L,LOLD)=U(L,LNEW)                                               MXFLUT12
      HCINCP(L,LOLD)= HCINCP(L,LNEW)                                    MXFLUT13
      ENTH(JRICH,L,LOLD)= ENTH(JRICH,L,LNEW)                            MXFLUT14
      ENTH(JLEAN,L,LOLD)= ENTH(JLEAN,L,LNEW)                            MXFLUT15
C*  MOVE CONCENTRATIONS FROM NEW TO OLD LOCATIONS UNLESS YOU HAVE JUST CMXFLUT16
C*  --SCKP/GCKP--                                                       MXFLUT17
      DO 298 I=1,NCONC                                                  MXFLUT18
      CONC(I,JRICH,L,LOLD)=CONC(I,JRICH,L,LNEW)                         MXFLUT19
  298 CONC(I,JLEAN,L,LOLD)=CONC(I,JLEAN,L,LNEW)                         MXFLUT20
  299 CONTINUE                                                          MXFLUT21
      IF( PDUM(13).NE.0. ) CALL TABPRT(6HHCINCP,HCINCP,24,10)           MXFLUT22
      IF( PDUM(13).NE.0. ) WRITE (6,2880) (CONC(16,1,K,2),K=1,12)       MXFLUT23
 2880 FORMAT(/2X,5HFUEL2//2X,6E16.8/2X,6E16.8)                          MXFLUT24
      RETURN                                                            MXFLUT25
    1 FORMAT (1X,4HTUBE,I3,2X,12F10.6)                                  MXFLUT26
    2 FORMAT (1X,//)                                                    MXFLUT27
    3 FORMAT (1X,7HTO TUBE,I3,1X,12F10.6)                               MXFLUT28
    4 FORMAT (10X,12(10H  SPECIE  ))                                    MXFLUT29
    5 FORMAT (10X,12(I5,5X))                                            MXFLUT30
    6 FORMAT (10X,49HFLOW OF EACH SPECIE IN EACH TUBE AT OLD X-STATION,/MXFLUT31
     1,35X,10H**XFRACT**)                                               MXFLUT32
    7 FORMAT (10X,49HFLOW OF EACH SPECIE IN EACH TUBE AT NEW X-STATION,/MXFLUT33
     1,35X,10H**XFRHAT**)                                               MXFLUT34
    8 FORMAT (10X,36HSOLUTION FOR INTERTUBE MASS TRANSFER,/,35X,9H**ALPHMXFLUT35
     1A**)                                                              MXFLUT36
    9 FORMAT (12X,12(10H   FROM   ),/,12X,12(10H   TUBE   ),/,12X,12(I5,MXFLUT37
     15X))                                                              MXFLUT38
   10 FORMAT (1X,35HMEAN FUEL TO GAS RATIO IN EACH TUBE,/,1X,12F10.6,/) MXFLUT39
      END                                                               MXFLUT40
```

```
CMXKINO         MAIN SUB.-- MIX. FLUTTER, KINETICS, OUTPUT
      SUBROUTINE MXKINO
      LOGICAL         LKGCKP,LKSCKP
      COMMON /INDATA/ N,HF,WAR,T2,BETA,T25,FAR5,FINO2C,PO,
     *                RCO(11),RCO2(11),RHC(11),RNOX(11),
     *                PT(11),PS(11),BLOC(11),QCO(11)
      REAL            N
      COMMON /JETDAT/ NPTS,RAD(12),TS(12),UU(12),SPV(12),MWT(12),
     *                CP(12),FUU (12),SPALDG(12),TKE(12),OTHER1(36)
      REAL            MWT
      COMMON /GASCMP/ RICH(12,2),FUEL(2,12,2),ENTH(2,12,2),
     *                CONC(16,2,12,2),HCINCP(12,2),OTHER(192)
      COMMON /GASTMW/ TG(2,12,2),MWTG(2,12,2),TAU(12,2),CPG(2,12,2)
      REAL            MWTG
      COMMON /OPCTRL/ TITLE(20),PRINT(30)
      COMMON /STCTRL/ LSTA,FINAL,CHEMK,FIRSTM,FIRSTC,SC,DXC,NIST,
     *                DUMST(9)
      LOGICAL         FINAL,FIRSTM,FIRSTC
      INTEGER         CHEMK
      COMMON /CPRINT/ PDUM(20)
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
      LOGICAL         ERR,ERRMAJ,INERR,PRERR
      COMMON /CBITS / BITS, BLANK
      COMMON /CINSAV/ HGCKP(57)
      COMMON /KOUT/ TITLE1(20),DUMKT(37)
      COMMON /CMASS / DUMASS(48)
      COMMON /CAXIAL/ DUMAX(3)
      COMMON /PORIDE/ PSTA(30)
      COMMON /OUTREM/ KINFIL
      COMMON /COREQN/ COREQ
      LOGICAL         COREQ
      DIMENSION CLASS(2)
C
      NAMELIST/A/ PDUM,CHEMK,PSTA,NIST ,COREQ
C
C     CALCULATE FUEL MW AND ADJUST C2H4 THERMO COEFF
C
      CALL ADJMC
C
C     INITIALIZE
C
      KINFIL=40
      CALL SETM(1,BITS,PSTA,30,)
      CALL SETM(3,0.,PDUM,20,HCINCP,24,TAU,24)
      NIST   = 2
      CHEMK  = 1
      XC     = 0.
      LSTA   = 1
      FIRSTM= .TRUE.
      FIRSTC= .TRUE.
```

```
      CALL ERRORW( $300,$300 )
      READ (5,A)
C
C     CALCULATE MIXTURE FLUTTER
C
    1 CALL LLINK( 4HLMXF )
      LKGCKP= .FALSE.
      LKSCKP= .FALSE.
      CALL MFMAIN
      CALL MOVE(1,TITLE,TITLE1,20,1)
      FIRSTM= .FALSE.
      NK     = NPTS
C
C     MIXTURE FLUTTER DATA STORED AT PRESENT (2) STATION
C     CALCULATE MOLECULAR WEIGHTS
C
      CALL MOVE(2,MWTG(1,1,2),MWTG(1,1,1),24,1,TG(1,1,2),
     * TG(1,1,1),24,1)
      DO 6 K=1,NK
      DO 5 J=1,2
C
C     RESET N ATOM COMPOSITION
C
      IF( CONC(12,J,K,2).LE. 1.E-8 ) CONC(12,J,K,2)=0.
      SUMC  = 0.
      DO 4 I=1,16
    4 SUMC  = SUMC+CONC(I,J,K,2)
    5 MWTG(J,K,2)= 1./SUMC
    6 CONTINUE
C
C     CALCULATE STATIC TEMPERATURE FROM ENTHALPY
C
   10 CALL TFMH( NK )
C
      IF( PDUM(14).EQ.0. ) GO TO 15
      CALL TABPRT(6HGASCMP,RICH,912,10)
      CALL TABPRT(6HGASTMW,TG,96,10)
   15 CONTINUE
C
C*** SECTION TO CALCULATE TIME HISTORY OF REACTIONS
C
      DO 60 K=1,NK
C
C    RECORD STARTING POINT FOR KINETICS STEP (JUST IN CASE)
C
      REWIND KINFIL
      DATA CLASS /4HRICH,4HLEAN/
      WRITE(KINFIL,1933)K,RICH(K,2),HCINCP(K,2),(TAU(K,L),L=1,2),
     1(CLASS(L),FUEL(L,K,2),TG(L,K,2),ENTH(L,K,2),(CONC(IC,L,K,2),
     1IC=1,16),L=1,2)
```

```
 1933 FORMAT(49H INITIAL CONDITIONS FOR LAST KINETICS CALCULATION,
     160X,11HSTREAMTUBE ,I2,10X/132X/8H YRICH= ,F11.8,10H  HCINCP= ,
     1F11.8,12H  TIME FROM ,E15.8,4H TO ,E15.8,8H SECONDS,38X/(132X/
     11X,A4,14H PART   FUEL= ,F12.9,5H  T= ,F10.4,5H  H= ,E15.8,
     128H   GAS COMPOSITION, MOLES/LB,38X/1X,8E15.8,11X/1X,8E15.8,11X))
C
   30 GO TO (40,45,50) , CHEMK
   50 CALL CHEMB( IBRNCH )
C
      GO TO (40,45) , IBRNCH
C
   40 IF( LKGCKP ) GO TO 41
      CALL LLINK( 4HLGCK )
      LKGCKP= .TRUE.
      LKSCKP= .FALSE.
   41 CALL GCKP( K )
      IF( ERR ) GO TO 205
      GO TO 60
C
   45 IF( LKSCKP ) GO TO 46
      CALL LLINK( 4HLSCK )
      LKSCKP= .TRUE.
      LKGCKP= .FALSE.
   46 CALL SCKP( K )
      IF( ERR ) GO TO 200
   60 CONTINUE
      CALL MOVE(1,TAU(1,2),TAU(1,1),12,1)
C
C     TEST FOR PRINT STATION
C
      IF( .NOT.FINAL ) GO TO 1
      CALL LLINK(4HLFIN)
      LKSCKP=.FALSE.
      LKGCKP=.FALSE.
      CALL SUMUP
      FINAL = .FALSE.
      LSTA  = LSTA+1
      IF( PRINT(LSTA) .NE.BITS ) GO TO 1
  100 RETURN
C
  200 WRITE (6,201)
  201 FORMAT(//2X,25HERROR IN SCKP CALCULATION//)
      GO TO 300
  205 WRITE (6,206)
  206 FORMAT(//2X,25HERROR IN GCKP CALCULATION//)
  300 ERR   = .TRUE.
      IF(.NOT.LKSCKP) CALL LLINK(4HLSCK)
      CALL ERROR1
      RETURN
      END
```

```
CNAMBLK     *NAMBLK*  BLOCK DATA      ( A/N )
      BLOCK DATA
C
C     ALPHANUMERIC DATA FOR TESTING AND OUTPUT
C
*     *** FOR GCKP=1      AUGUST 1973 *****
C
      COMMON / LTUS / LTHM, LDAT
C
      COMMON / OPTS / TIMIN, TIME, BLNK, AREA, DUM3(4)
C
      COMMON / SPEC / SNAM(2,3), DUM4(2,25), EFFM(2), BLANK(2),
     * DUM5(25,62)
C
      COMMON / KOUT / TITLE(20), UNITI, UNITO, DUM6(32), FPS, SI, DUM7
C
*     LOGICAL TAPE UNIT ASSIGNMENTS
C
C     *** ( NOT USED IN GCKP=1 )***
      DATA     LTHM, LDAT/4, 43/
C
*     ALPHANUMERIC DATA
C
      DATA     TIME, AREA/4HTIME,4HAREA/
      DATA     SNAM, EFFM, BLANK/1HV,1H ,3HRHO,1H ,1HT,1H ,1HM,1H ,
     1 1H ,1H /
      DATA     FPS, SI/3HFPS,2HSI/
C
      DATA     TITLE/
     * 4H    , 4H    , 4HKASK, 4HANS , 4HEXPE, 4HRIME,
     * 4HNT -, 4H POS, 4HT-FL, 4HAME , 4HREAC, 4HTION,
     * 4HS IN, 4H LEA, 4HN C2, 4HH4-A, 4HIR F, 4HLAME,
     * 4H    , 4H    /
      DATA     UNITI,UNITO/3HFPS,3HFPS/
      DATA     BLNK/1H /
      DATA     TIMIN/4HTIME/
C
      END
```

```
CNOX2B          INTEGRATE NO RATE EQUATIONS
      SUBROUTINE NOX2B( P,DTIME,XNOI )
      COMMON /PSEQ  / FOA,BETS,TP,X(16),DHQDMW,TEQ,BEQ,XMWT,HNEQ
      EQUIVALENCE (X1,X(1)),(X2,X(2)),(X3,X(3)),(X4,X(4)),(X5,X(5)),
     *            (X7,X(7)),(X8,X(8)),(X9,X(9)),(X10,X(10)),
     *            (X11,X(11)),(X12,X(12))
      COMMON /GHSC  /  F(25),HH(25),SR(25),CPZ(25),DCPR(25)
      COMMON /NOXTRA/ DNOXDT
      REAL      MWT,K1,K2,K3,KR1,KR2,KR3
      DIMENSION RK(4)
      DATA C2/.01603286/,K3/40.E+12/
C
C
      TK     = TP / 1.8
      RHOM   = 144.*P*C2/(1545.32*TP)
      N      = 1
C
C     DETERMINE EQUILIBRIUM CONSTANTS
C
    1 CALL THRM( TK,1. )
      EQ1    = EXP(F(9)+F(2)-F(11)-F(12))
      EQ2    = EXP(F(12)+F(4)-F(11)-F(2))
      EQ3    = EXP(F(12)+F(5)-F(11)-F(1))
      K1     = EQ1*3.10E13*EXP(-168.1874/TK)
      K2     = 6.40E09*TK*EXP(-3147.22/TK)
      KR1    = K1/EQ1
      KR2    = K2/EQ2
      KR3    = K3/EQ3
      TIMEK  = DTIME
C
C     COMPUTE DXNO/DT FOR RK INTEGRATION
C
      XNOT   = XNOI
    5 R1     = 2.*K1*RHOM*(X9*X2*(K2*X4+K3*X5)-XNOT**2/EQ1*
     * (KR2*X2+KR3*X1))
      R2     = KR1*XNOT+K2*X4+K3*X5
C
   15 RK(N)  = R1/R2
      GO TO (16,16,20,40) , N
   16 XNOT   = XNOI+.5*RK(N)*TIMEK
      GO TO 25
   20 XNOT   = XNOI+RK(N)*TIMEK
   25 N      = N+1
      GO TO 5
   40 DNOXDT= (RK(1)+2.*(RK(2)+RK(3))+RK(4))/6.
      XNOI   = XNOI+DNOXDT*TIMEK
      X12    = (K1*X9*X2+KR2*XNOT*X2+KR3*XNOT*X1)/R2
C
C
  100 RETURN
```

END NOX28051

```
CNWNET        CONVERT JETMIX PROFILES TO FIXED-NUMBER STREAM NET
 C            SUITABLE FOR MIXING-REACTION CALCULATIONS
 C
       SUBROUTINE NEWNET
 C
       INTEGER AGAIN
       REAL MWTJX,MWTO
       DIMENSION GJX(50),WKCJX(50,12),UDF(12)
 C
       COMMON /CAGAIN/ AGAIN
       COMMON/CINPJT/DIAJ,DUM3(64)
       COMMON/JETDAT/N,RAD(12),TO(12),U(12),SPVO(12),MWTO(12),CPO(12),
      'FO(12),G(12),TKEO(12),OTHRO(36)
       COMMON/CCHECK/DUM4(12),WKTOT(12),DUM5(12)
       COMMON/CNWNET/WFCJX(50),DUM6(50),WFCUM(12),DUM7(50),PSIJX(50)
       COMMON/CSPECI/M,NF,DX
       COMMON/CINPUT/XX(5),RJR(5,50),UJR(5,50),RHOJR(5,50),
      'XKJR(5,50,13),PSIJR(5,50),GJR(5,50),NJJ(5),NII
       COMMON/CAXIAL/DUM8,DUM9,X
       COMMON/CLOCAL/RJX(50),UJX(50),DUM10(50),XKJX(50,13),NJ
       COMMON/CMXEL1/W(12,12),WHAT(13,12),ALPHA(12,12)
       COMMON/CMASS/WTOLD(12),WTOT(12),DUM11(24)
       COMMON/GASTMW/DUM1(96),TAU(12,2),DUM2(48)
       COMMON /CPRINT/ PDUM(20)
       DIMENSION CFACSP(13)
       EQUIVALENCE (CORFAC,CFACSP(13))
 C
 C    FIRST TIME THRU, DERIVE CONTINUITY CHECK PARAMETERS
 C
       IF(AGAIN.EQ.(-1))GO TO 12
 C
 C    REPEAT PREJET CALCULATION OF INITIAL TUBE AREAS
 C
 C     SET UP SPECIE FLOW MATRIX FOR INITIAL STEP PROFILE
 C
       CALL SETM(2,0.,WHAT,156,WTOT,12)
       RTI2=0.0
       WFCUM(1)=0.0
       DO 10 J=1,N
       RTO2=0.5*(RAD(J)**2+RAD(J+1)**2)
       IF(J.EQ.N)RTO2=RAD(M)**2
       ATO=3.14156*(RTO2-RTI2)
       RTI2=RTO2
 C
 C    TOTAL FLOW OF EACH SPECIE AND
 C    CUMULATIVE FUEL FLOW IN EACH TUBE.
 C
       WKTOT(J)=ATO*U(J)/SPVO(J)
       WTOT(J)= WKTOT(J)
       WHAT(J,J)= WKTOT(J)
```

```
      WF=FO(J)*WKTOT(J)
      WFCUM(J+1)=WFCUM(J)+WF
   10 CONTINUE
C
C  SET SPECIES FLOW MATRIX AT OLD STATION EQUAL TO PREVIOUSLY
C  DERIVED FLOW MATRIX
C
   12 DO 14 J=1,M
   14 CALL MOVE(1,WHAT(1,J),W(1,J),12,1)
C
C  STORE TOTAL FLOW IN EACH TUBE AT OLD STATION
C
      CALL MOVE(1,WTOT,WTOLD,12,1)
C
C  INTERPOLATE WITH AXIAL DISTANCE IN JETMIX PROPERTY TABLE
C  FROM READT TO ESTABLISH JETMIX PROFILE AT X.
C  FIRST,RADIUS (RJX) AND VELOCITY (UJX)
C
   20 NJ=NJJ(1)
      DO 50 JJ=1,NJ
      CALL LSPFIT(XX,RJR(1,JJ),NII,X,RJX(JJ),1,0)
      CALL LSPFIT(XX,UJR(1,JJ),NII,X,UJX(JJ),1,0)
C
C  GET MOLE FRACTION OF SPECIE K AT POINT JJ (XKJX) AND
C  MEAN MOLECULAR WEIGHT
C
      MWTJX=0.0
      DO 30 K=1,M
      CALL LSPFIT(XX,XKJR(1,JJ,K),NII,X,XKJX(JJ,K),1,0)
   30 MWTJX=MWTJX+XKJX(JJ,K)*MWTO(K)
C
C  CONVERT MOLE FRACTIONS TO MASS FRACTIONS AND
C  COMPUTE FUEL MASS FRACTION (STORE IN XKJX(JJ,M+1)
C
      XKJX(JJ,NF)=0.0
      DO 40 K=1,M
      XKJX(JJ,K)=XKJX(JJ,K)*MWTO(K)/MWTJX
   40 XKJX(JJ,NF)=XKJX(JJ,NF)+XKJX(JJ,K)*FO(K)
C
C  GET SPALDING HETEROGENIETY FACTOR GJX
C
      CALL LFIT(XX,GJR(1,JJ),NIT,X,GJX(JJ),1,0)
   50 CONTINUE
C
C   GET VALUE OF STREAM FUNCTION (PSIJX) AT POINT JJ, THEN
C  CUMULATIVE SPECIE FLOWS (WKCJX) AND FUEL FLOW (WFCJX)
C
      PSIJX(1)=0.0
      WFCJX(1)=0.0
      DO 55 K=1,M
```

```
   55 WKCJX(1,K)=0.0
      DO 65 JJ=2,NJ
      CALL LFIT(XX,PSIJR(1,JJ),NTI,X,PSIJX(JJ),1,0)
      DWJX=PSIJX(JJ)-PSIJX(JJ-1)
      DO 60 K=1,M
   60 WKCJX(JJ,K)=WKCJX(JJ-1,K)+0.5*(XKJX(JJ-1,K)+XKJX(JJ,K))*DWJX
   65 WFCJX(JJ)=WFCJX(JJ-1)+0.5*(XKJX(JJ-1,NF)+XKJX(JJ,NF))*DWJX
C
C   SCALE FUEL FLOWS TO MATCH JET EXIT
C
      CORFAC=WFCUM(M)/WFCJX(NJ)
      DO 70 JJ=1,NJ
   70 WFCJX(JJ)=WFCJX(JJ)*CORFAC
C
C   TABULATE RADIUS AND SPECIE FLOWS AGAINST FUEL FLOW.
C   INTERPOLATE TO GET REACTION TUBE BOUNDARIES AND SPECIE FLOWS.
C
      CALL LSPFIT(WFCJX,RJX,NJ,WFCUM,RAD,M,0)
      DO 75 K=1,M
   75 RAD(K)= SQRT( RAD(K) )
      DO 90 K=1,M
      CALL LSPFIT(WFCJX,WKCJX(1,K),NJ,WFCUM,WHAT(1,K),M,0)
C
C   CONVERT CUMULATIVE SPECIE FLOWS TO INCREMENTAL FLOWS IN EACH TUBE
C   INVERT SPECIE FLOW MATRIX USING ALPHA FOR TEMP. STORAGE.
C
      DO 80 J=1,N
      WHAT(J,K)=WHAT(J+1,K)-WHAT(J,K)
   80 ALPHA(K,J)=WHAT(J,K)
   90 CONTINUE
      DO 100 J=1,N
  100 CALL MOVE(1,ALPHA(1,J),WHAT(1,J),12,1)
C
C   INTEGRATE VELOCITY AND HETEROGENEITY ACROSS EACH REACTION TUBE
C   TO GET MEAN VALUE.
C
      UDF(1)= UJX(1)
      G(1)  = GJX(1)
      CALL LFIT(WFCJX,UJX,NJ,WFCUM,UDF,M,-1)
      CALL LFIT(WFCJX,GJX,NJ,WFCUM,G,M,-1)
      DO 110 J=1,N
      DWF=WFCUM(J+1)-WFCUM(J)
      U(J)=(UDF(J+1)-UDF(J))/DWF
  110 G(J)=AMAX1(0.0,(G(J+1)-G(J))/DWF)
      G(M) = 0.
C
C   CORRECT SPECIE FLOWS IN EACH TUBE SO THAT TOTAL FLOW OF EACH SPECIE
C   AGREES WITH INITIAL VALUE. AIR FLOW LEFT AS IS.
C
      DO 140 K=1,N
```

```
      WHAT(K,M)=0.0
      SUMWK=0.0
      DO 120 J=1,M
  120 SUMWK=SUMWK+WHAT(K,J)
      CFACSP(K)= WKTOT(K)/SUMWK
      DO 130 J=1,M
  130 WHAT(K,J)= WHAT(K,J)*CFACSP(K)
  140 CONTINUE
C
C   GET TOTAL FLOW IN EACH TUBE (WTOT).
C
      WHAT(M,M)=0.0
      WTOT(M)= 0.
      DO 151 J=1,N
      WTOT(J)=0.0
      DO 150 K=1,M
  150 WTOT(J)=WTOT(J)+WHAT(K,J)
  151 WTOT(M)= WTOT(M)+WTOT(J)
C   DEFINE OUTER TUBE FOR ENTRAINED AIR (W IS SPECIE FLOW
C   MATRIX AT OLD STATION).
C
      W(M,M)=0.0
      DO 160 J=1,N
  160 W(M,M)=W(M,M)+WHAT(M,J)-W(M,J)
      IF( PDUM(16).EQ.0. ) GO TO 152
      CALL TABPRT(6HCFACSP,CFACSP,13,10)
      CALL TABPRT(5HPSIJR,PSIJR,250,10)
      CALL TABPRT(5HPSIJX,PSIJX,NJ,10)
      CALL TABPRT(4HRJX2,RJX,NJ,10)
      CALL TABPRT(3HRAD,RAD,M,10)
      CALL TABPRT(5HWFCUM,WFCUM,M,10)
      CALL TABPRT(4HWTOT,WTOT,M,10)
C
  152 IF( AGAIN.NE. (-1) ) GO TO 180
C
C   REACTION TIME (TAU) UP TO NEW STATION (SECONDS).
C
      DXFT=DX*DIAJ/12.0
      DO 170 J=1,N
  170 TAU(J,2)=TAU(J,2)+DXFT/U(J)
C
  180 AGAIN=-1
      RETURN
      END
```

```
CNWPSI       REDISTRIBUTION OF STREAM FUNCTION FROM JETMIX
C            BASED ON LAST KNOWN FLOWS IN NEWNET TUBES.
C
      SUBROUTINE NEWPSI(PSINEW,NN)
C
      DIMENSION PSINEW(50)
      COMMON/CMASS/DUM1(12),FLOW(12),DUM2(24)
      COMMON/JETDAT/NT,RAD(12),T(12),U(12),SPV(12),DUM3(96)
C
C   BASIC NUMBER OF PSI'S PER NEWNET TUBE AND LAST TUBE
C   NOT TO RECEIVE EXTRA PSI
C
      NPTB=(NN-1)/NT
      NTX1=NT-MOD((NN-1),NT)
C
C   FIRST PSI IS ALWAYS CENTERLINE
C
      N=1
      PSINEW(N)=0.0
C
C   STEP FROM TUBE TO TUBE
C
      DO 20 I=1,NT
C
C   FRACTION OF FLOW IN THIS TUBE
C
      DPSI=FLOW(I)/FLOW(NT+1)
C
C   NUMBER OF PSI'S HERE AND INCREMENT
C
      NPT=NPTB
      IF(I.GT.NTX1)NPT=NPTB+1
      DPSI=DPSI/FLOAT(NPT)
C
C   EVALUATE PSI'S IN THIS TUBE
C
      DO 10 J=1,NPT
      N=N+1
   10 PSINEW(N)=PSINEW(N-1)+DPSI
   20 CONTINUE
      RETURN
C
C   IN THE BEGINNING, THERE ARE NO FLOWS
C
      ENTRY NEWFLO
      NT1=NT+1
      RTI2=0.0
      FLOW(NT1)=0.0
      DO 30 I=1,NT
      RTO2=0.5*(RAD(I)**2+RAD(I+1)**2)
```

```
      IF(I.EQ.NT1)RTO2=RAD(NT1)**2                                    NWPSI051
      FLOW(I)=3.14159*(RTO2-RTI2)*U(I)/SPV(I)                         NWPSI052
      FLOW(NT1)=FLOW(NT1)+FLOW(I)                                     NWPSI053
   30 RTI2=RTO2                                                       NWPSI054
      RETURN                                                          NWPSI055
      END                                                             NWPSI056
```

```
COUTPP      GENERAL OUTPUT, UNIT CONVERSIONS, ETC
      SUBROUTINE OUTP
C
C     OUTPUT CAN BE GIVEN IN  (1) INTERNAL (CGS) UNITS, (2) FPS UNITS
C      (3) SI UNITS
C
      LOGICAL ALLM1,CONC,DBUGO,FXCHR,NEXT,RHOCON,TCON,BRIEF
C
      REAL MDOT,TVAR,N,M,MW,MIXMW,M2,MACH,ISUBM
C
      DIMENSION SPNM(2,27),PRC(25),PRX(30),XXH(30),ESP(2,25),DUM1(41)
C
      COMMON/OPTS/VERSI,TIMEV,VERSA,AREAV,FLIM,TCON,RHOCON,IPRCOD
      COMMON/COND/DVAR,AREA,MDOT,P,IVAR,V,RHO,T,SIGMA(25),LS,LSP3,NEXT
      COMMON/SINT/HMIN,HINI,HN,HNP1,HMAX,NH,AVH,EMAX,ERRN,JCV,KOUNT,EPRP
      COMMON/KOUT/TITLE(20),UNITI,UNITO,CONC,FXCHR,DELH(30),FPS,SI,DBUGD
      COMMON/REAC/LSR(4,30),XX(30),RATE(30),LKEQ(30),DLKEQ(30),MM(30),LR
      COMMON/RRAT/A(30),N(30),FACT(30),B(30),M(25,30),ALLM1
      COMMON/AFUN/CN(4),ITPSZ,LSUBM,ETA,D,VISC,BETAL
      COMMON/SPEC/SNAM(2,50),MW(25),W(25),STOIC(25,30),OMEGA(25,30)
      COMMON/XVSA/XTB(40),ATB(40),NTB,XU,AU(2),CX(4),BRIEF,TIMLMT
      COMMON/NECC/RR,MIXMW,M2,GAMMA,TCPR,R
      COMMON/PORE/PK(28),QK(28),RK(28),E(28)
      COMMON/DFRN/F(28),ALPHA(28),BETA(28,28)
      COMMON/SKIP/NEGL(25),I1,I2,IT
      COMMON/GHSC/GRT(25),HRT(25),SR(25),CPR(25),DCPR(25)
      COMMON/SABS/S1,AA,BB,S2,DA,D2A,DTERM,RHD
C
      EQUIVALENCE (SPNM,SNAM(1,4)),(BLANK,SPNM(1,27))
      EQUIVALENCE (PRX(1),XXH(1)),(DUM1(1),ESP(1,1))
C
      ENTRY OUT1
C
C ** TITLE PAGE
      IF (VERSI .EQ. TIMEV) GO TO 98
      I = 2
      GO TO 99
   98 I = 4
   99 IF (VERSA .EQ. AREAV) I = I - 1
      GO TO (100,200,300,400),I
  100 WRITE (6,101)
  101 FORMAT (1H1,14X,21HDISTANCE-AREA VERSION)
      GO TO 3
  200 WRITE (6,201)
  201 FORMAT (1H1,12X,25HDISTANCE-PRESSURE VERSION)
      GO TO 3
  300 WRITE (6,301)
  301 FORMAT (1H1,16X,17HTIME-AREA VERSION)
      GO TO 3
  400 WRITE (6,401)
```

```
  401 FORMAT (1H1,14X,21HTIME-PRESSURE VERSION)
    3 WRITE (6,102) (TITLE(I),I=1,20)
  102 FORMAT (1H+,49X,33HGENERAL CHEMICAL KINETICS PROGRAM,11X,26HNASA L
     *EWIS RESEARCH CENTER///26X,20A4///9X,8HREACTION,31X,8HREACTION,
     * 38X,23HREACTION RATE VARIABLES/10X,6HNUMBER,74X,1HA,16X,1HN,9X,
     * 10HACTIVATION/119X,6HENERGY)
      KEY=1
      IF(BRIEF)WRITE(3)KEY,TITLE,UNITO,FPS,SI,LS,SPNM
C
C     PRINT REACTION INFORMATION
      DO 6 J=1,LR
      N1 = LSR(1,J)
      N2 = LSR(2,J)
      N3 = LSR(3,J)
      N4 = LSR(4,J)
      WRITE (6,103) J,(SPNM(I,N2),I=1,2),(SPNM(I,N3),I=1,2),A(J),N(J),
     * EACT(J)
  103 FORMAT (12X,I2,27X,2A4,2X,1H=,2X,2A4,23X,1PE12.5,5X,0PF10.4,5X,
     * F10.2)
      IF (N1 .GT. 0) GO TO 5
      IF (N1 .LT. 0) GO TO 4
      N1 = 26
      GO TO 205
    4 WRITE (6,105) (SPNM(I,N4),I=1,2)
  105 FORMAT (1H+,63X,1H+,2X,2A4)
      GO TO 6
    5 IF (N4 .GT. 0) GO TO 205
      IF (N4 .LT. 0) GO TO 204
      N4 = 26
      GO TO 205
  204 WRITE (6,1105) (SPNM(I,N1),I=1,2)
 1105 FORMAT (1H+,27X,2A4,2X,1H+)
      GO TO 6
  205 WRITE (6,104) (SPNM(I,N1),I=1,2),(SPNM(I,N4),I=1,2)
  104 FORMAT (1H+,27X,2A4,2X,1H+,25X,1H+,2X,2A4)
C
C     CONVERT ACTIVATION ENERGY TO B-FACTOR
    6 B(J) = EACT(J)/1.987165
C
      IF (.NOT. ALLM1) GO TO 7
      WRITE (6,106)
  106 FORMAT (///51X,29HALL THIRD BODY RATIOS ARE 1.0)
      GO TO 13
C
    7 WRITE (6,107)
  107 FORMAT (///41X,50HALL THIRD BODY RATIOS ARE 1.0 EXCEPT THE FOLLOWI
     *NG//)
      K = 0
      DO 12 I=1,LS
      DO 12 J=1,LR
```

```
      IF (M(I,J) .EQ. 1.) GO TO 12
      K = K + 1
      IF (K .EQ. 5) K = 1
      GO TO (8,9,10,11),K
    8 WRITE (6,108) (SPNM(K,I),K=1,2),J,M(I,J)
  108 FORMAT (5X,2HM(,2A4,1H,,I2,3H) =,F10.5)
      GO TO 12
    9 WRITE (6,109) (SPNM(K,I),K=1,2),J,M(I,J)
  109 FORMAT (1H+,36X,2HM(,2A4,1H,,I2,3H) =,F10.5)
      GO TO 12
   10 WRITE (6,110) (SPNM(K,I),K=1,2),J,M(I,J)
  110 FORMAT (1H+,68X,2HM(,2A4,1H,,I2,3H) =,F10.5)
      GO TO 12
   11 WRITE (6,111) (SPNM(K,I),K=1,2),J,M(I,J)
  111 FORMAT (1H+,100X,2HM(,2A4,1H,,I2,3H) =,F10.5)
   12 CONTINUE
C
   13 IF (VERSI .EQ. TIMEV) GO TO 14
      WRITE (6,112) HMIN,HMAX,HINT,EMAX
  112 FORMAT (///56X,20HINTEGRATION CONTROLS//15X,17HMINIMUM STEP SIZE,
     * E14.5,3H CM,33X,17HMAXIMUM STEP SIZE,E14.5,3H CM//15X,17HINITIAL
     *STEP SIZE,E14.5,3H CM,33X,22HMAXIMUM RELATIVE ERROR,F10.5)
      GO TO 15
   14 WRITE (6,113) HMIN,HMAX,HINT,EMAX
  113 FORMAT (///56X,20HINTEGRATION CONTROLS//15X,17HMINIMUM STEP SIZE,
     * E14.5,4H SEC,32X,17HMAXIMUM STEP SIZE,E14.5,4H SEC//15X,17HINITIA
     *L STEP SIZE,E14.5,4H SEC,32X,22HMAXIMUM RELATIVE ERROR,F10.5)
C
C ** SECOND PAGE
   15 WRITE (6,114)
  114 FORMAT (1H1,50X,31H** ASSIGNED VARIABLE PROFILE **//)
      GO TO (16,18,19,19,20),ITPSZ
C
   16 GO TO (116,216,316,416),IPRCOD
C     ASSIGNED VARIABLE TABLE
  116 WRITE (6,117) XU,AU
  117 FORMAT (34X,64HTHE AREA IS CALCULATED BY INTERPOLATION FROM THE FO
     *LLOWING TABLE//36X,7HSTATION,10X,17HAXIAL DISTANCE  (,A2,1H),10X,
     * 7HAREA  (,A4,A1,1H))
      GO TO 516
  216 WRITE (6,217) XU,AU
  217 FORMAT (32X,68HTHE PRESSURE IS CALCULATED BY INTERPOLATION FROM TH
     *E FOLLOWING TABLE//36X,7HSTATION,10X,17HAXIAL DISTANCE  (,A2,1H),
     * 9X,11HPRESSURE  (,2A4,1H))
      GO TO 516
  316 WRITE (6,317) AU
  317 FORMAT (34X,64HTHE AREA IS CALCULATED BY INTERPOLATION FROM THE FO
     *LLOWING TABLE//36X,7HSTATION,14X,11HTIME  (SEC),16X,7HAREA  (,A4,
     * A1,1H))
      GO TO 516
```

```
  416 WRITE (6,417) AU
  417 FORMAT (32X,68HTHE PRESSURE IS CALCULATED BY INTERPOLATION FROM TH
     *E FOLLOWING TABLE//36X,7HSTATION,14X,11HTIME  (SEC),15X,11HPRESSUR
     *E  (,2A4,1H))
  516 DO 17 I=1,NTB
   17 WRITE (6,616) I,XTB(I),ATB(I)
  616 FORMAT (38X,I2,14X,1PE12.5,15X,E12.5)
      GO TO 21
C
   18 GO TO (218,318,418,518),IPRCOD
C     ASSIGNED VARIABLE POLYNOMIAL
  218 WRITE (6,219) AU,CX
  219 FORMAT (40X,52HTHE AREA IS CALCULATED FROM THE FOLLOWING POLYNOMIA
     *L//23X,6HAREA (,A4,A1,5H) = (,1PE12.5,9H)X**3 + (,E12.5,9H)X**2 +
     *(,E12.5,6H)X + (,E12.5,1H))
      GO TO 21
  318 WRITE (6,319) AU,CX
  319 FORMAT (38X,56HTHE PRESSURE IS CALCULATED FROM THE FOLLOWING POLYN
     *OMIAL//20X,10HPRESSURE (,2A4,5H) = (,1PE12.5,9H)X**3 + (,E12.5,9H)
     *X**2 + (,E12.5,6H)X + (,E12.5,1H))
      GO TO 21
  418 WRITE (6,419) AU,CX
  419 FORMAT (40X,52HTHE AREA IS CALCULATED FROM THE FOLLOWING POLYNOMIA
     *L//23X,6HAREA (,A4,A1,5H) = (,1PE12.5,9H)T**3 + (,E12.5,9H)T**2 +
     *(,E12.5,6H)T + (,E12.5,1H))
      GO TO 21
  518 WRITE (6,519) AU,CX
  519 FORMAT (38X,56HTHE PRESSURE IS CALCULATED FROM THE FOLLOWING POLYN
     *OMIAL//20X,10HPRESSURE (,2A4,5H) = (,1PE12.5,9H)T**3 + (,E12.5,9H)
     *T**2 + (,E12.5,6H)T + (,E12.5,1H))
      GO TO 21
C
C     SPECIAL AREA FUNCTION
   19 WRITE (6,118) LSUBM,ETA
  118 FORMAT (41X,50HTHE AREA IS CALCULATED FROM THE FOLLOWING FUNCTION/
     */46X,16H1/AREA = 1 - (X/,F10.3,4H)**(,F10.5,1H))
      IF (IIPSZ .EQ. 4) WRITE (6,1118) D,VISC,BETAL
 1118 FORMAT (/6X,20HHYDRAULIC DIAMETER =,F8.4,3H CM,7X,23HVISCOSITY COE
     *FFICIENT =,E12.4,10H GM/CM-SEC,7X,6HBETA =,F7.4)
      GO TO 21
C
C     ZERO VELOCITY - ASSIGNED VARIABLE NOT REQUIRED
   20 WRITE (6,119)
  119 FORMAT (36X,60HTHIS IS A V=0 PROBLEM - AN ASSIGNED VARIABLE IS NOT
     * REQUIRED)
C
C     NEGLECTED SPECIES
   21 IF (I1 .NE. 0) GO TO 22
      WRITE (6,120)
  120 FORMAT (///31X,70HNO SPECIES WILL BE PERMANENTLY NEGLECTED FROM AL
```

```
     *L ERROR CONSIDERATIONS)
      GO TO 228
   22 WRITE (6,121)
  121 FORMAT (///31X,69HTHE FOLLOWING SPECIES WILL BE NEGLECTED FROM ALL
     * ERROR CONSIDERATIONS//)
      K = 0
      DO 28 II=1,I1
      I = NEGL(II)
      K = K + 1
      IF (K .EQ. 6) K = 1
      GO TO (23,24,25,26,27),K
   23 WRITE (6,122) (SPNM(J,I),J=1,2)
  122 FORMAT (61X,2A4)
      GO TO 28
   24 WRITE (6,123) (SPNM(J,I),J=1,2)
  123 FORMAT (1H+,76X,2A4)
      GO TO 28
   25 WRITE (6,124) (SPNM(J,I),J=1,2)
  124 FORMAT (1H+,44X,2A4)
      GO TO 28
   26 WRITE (6,125) (SPNM(J,I),J=1,2)
  125 FORMAT (1H+,92X,2A4)
      GO TO 28
   27 WRITE (6,126) (SPNM(J,I),J=1,2)
  126 FORMAT (1H+,28X,2A4)
   28 CONTINUE
C
  228 IF (RHOCON) WRITE (6,1126)
 1126 FORMAT (///38X,56HTHE VOLUME (DENSITY) WILL BE HELD CONSTANT FOR T
     *HIS CASE)
      IF (TCON) WRITE (6,2126)
 2126 FORMAT (///40X,51HTHE TEMPERATURE WILL BE HELD CONSTANT FOR THIS C
     *ASE)
      RETURN
C
      ENTRY OUT2
C
C     INITIAL CONDITIONS
      KEY=2
      IF(.NOT.BRIEF)WRITE(6,127)
  127 FORMAT (1H1,52X,26H**  INITIAL CONDITIONS  **//)
      GO TO 29
C
      ENTRY OUT3
C
C     GENERAL OUTPUT
      KEY=3
      IF(NEXT.OR..NOT.BRIEF)WRITE(6,128)
  128 FORMAT (1H1)
C
```

```
   29 MACH = SQRT(M2)                                                    OUTPP251
      MAX = MAX0(LS,LR)                                                  OUTPP252
C                                                                        OUTPP253
      TENT = 0.                                                          OUTPP254
      CSUM = 0.                                                          OUTPP255
      PMLOG = ALOG(P*MIXMW)                                              OUTPP256
C     TOTAL ENTROPY AND MASS FRACTION SUM                                OUTPP257
      DO 30 I=1,LS                                                       OUTPP258
      IF (SIGMA(I) .EQ. 0.) GO TO 30                                     OUTPP259
      TENT = TENT + SIGMA(I)*(SR(I) - ALOG(SIGMA(I)) - PMLOG)            OUTPP260
   30 CSUM = CSUM + SIGMA(I)*MW(I)                                       OUTPP261
      TENT = TENT*1.987165                                               OUTPP262
C                                                                        OUTPP263
      TXXH = 0.                                                          OUTPP264
C     ENERGY EXCHANGE RATES                                              OUTPP265
      DO 31 J=1,LR                                                       OUTPP266
      XXH(J) = XX(J)*DELH(J)                                             OUTPP267
   31 TXXH = TXXH + XXH(J)                                               OUTPP268
C                                                                        OUTPP269
      IF (VERSI .EQ. TIMEV) GO TO 32                                     OUTPP270
      TIME = DVAR                                                        OUTPP271
      X = IVAR                                                           OUTPP272
      GO TO 33                                                           OUTPP273
   32 TIME = IVAR                                                        OUTPP274
      X = DVAR                                                           OUTPP275
C                                                                        OUTPP276
   33 IF(BRIEF.AND..NOT.NEXT)GO TO 90                                    OUTPP277
      IF(UNITD.NE.FPS)GO TO 48                                           OUTPP278
C                                                                        OUTPP279
C     CONVERT FROM INTERNAL (CGS) UNITS TO FPS UNITS                     OUTPP280
      X = X/30.48                                                        OUTPP281
      AREAA = AREA/929.0304                                              OUTPP282
      DOTM = MDOT/453.59237                                              OUTPP283
      PP = P*2116.2                                                      OUTPP284
      VV = V/30.48                                                       OUTPP285
      RHOO = RHO*62.43                                                   OUTPP286
      TT = T*1.8                                                         OUTPP287
C                                                                        OUTPP288
      WRITE (6,129) TIME,AREAA,X,PP,NH,VV,AVH,RHOO,(SNAM(I,JCV),I=1,2),  OUTPP289
     * TT,DOTM,TENT,FRRN,MACH,KOUNT,GAMMA                                OUTPP290
  129 FORMAT (16X,4HTIME,1PE14.5,5H  SEC,14X,4HAREA,E14.5,7H  SQ FT,     OUTPP291
     * 14X,14HAXIAL POSITION,E14.5,4H  FT///20X,15HFLOW PROPERTIES,45X,  OUTPP292
     *    22HINTEGRATION INDICATORS//22X,8HPRESSURE,E22.5,30X,21HSTEPS FROUTPP293
     *OM LAST PRINT,9X,I4/23X,10H(LB/FT**2)/22X,8HVELOCITY,E22.5,30X,    OUTPP294
     * 17HAVERAGE STEP SIZE,0PE24.5/23X,8H(FT/SEC)/22X,7HDENSITY,1PE23.5OUTPP295
     *,30X,20HCONTROLLING VARIABLE,11X,2A4/23X,10H(LB/FT**3)/22X,11HTEMPOUTPP296
     *ERATURE,E19.5/23X,7H(DEG R)/22X,14HMASS FLOW RATE,E16.5/23X,8H(LB/OUTPP297
     *SEC)/22X,7HENTROPY,E23.5,30X,14HRELATIVE ERROR,0PE27.5/23X,14H(BTUOUTPP298
     */LB/DEG R)/22X,11HMACH NUMBER,1PE19.5,30X,20HPREDICTOR ITERATIONS,OUTPP299
     * 11X,I3//22X,5HGAMMA,E25.5)                                        OUTPP300
```

```
C
      IF (I2 .EQ. 0) GO TO 34
      IS = I1 + 1
      DO 233 K=IS,IT
      KK = NEGL(K)
      ESP(1,K) = SPNM(1,KK)
  233 ESP(2,K) = SPNM(2,KK)
      WRITE (6,130) ((ESP(I,K),I=1,2),K=IS,IT)
  130 FORMAT (1H+,81X,18HELIMINATED SPECIES,13X,2A4,2X,2A4/(113X,2A4,2X,
     * 2A4))
C
   34 WRITE (6,131)
  131 FORMAT (//56X,19HCHEMICAL PROPERTIES//)
C
      CONV = 0.02883
      IF (CONC .OR. EXCHR) GO TO 36
C
C     PRINT MASS FRACTIONS AND REACTION CONVERSION RATES
      WRITE (6,132)
  132 FORMAT (1X,7HSPECIES,4X,13HMASS FRACTION,3X,13HMOLE FRACTION,3X,
     * 27HNET SPECIES PRODUCTION RATE,5X,8HREACTION,3X,28HNET REACTION C
     *ONVERSION RATE,3X,13HRATE CONSTANT)
      WRITE (6,133)
  133 FORMAT (50X,16H(MOLE/FT**3/SEC),11X,6HNUMBER,7X,22H(MOLE-FT**3/LB
     **2/SEC),7X,11H(CGS UNITS))
      DO 35 J=1,IR
   35 PRX(J) = XX(J)
      CONV = 1./62.43
      GO TO 37
C
   36 IF (CONC .OR. (.NOT. EXCHR)) GO TO 39
C
C     PRINT MASS FRACTIONS AND ENERGY EXCHANGE RATES
      WRITE (6,134)
  134 FORMAT (1X,7HSPECIES,4X,13HMASS FRACTION,3X,13HMOLE FRACTION,3X,
     * 27HNET SPECIES PRODUCTION RATE,5X,8HREACTION,5X,24HNET ENERGY EXC
     *HANGE RATE,5X,13HRATE CONSTANT)
      WRITE (6,135)
  135 FORMAT (50X,16H(MOLE/FT**3/SEC),11X,6HNUMBER,8X,21H(BTU-FT**3/LB**
     *2/SEC),7X,11H(CGS UNITS))
C
C     COMPUTE MASS FRACTIONS
   37 DO 38 I=1,IS
   38 PRC(I) = SIGMA(I)*MW(I)
      GO TO 44
C
   39 IF ((.NOT. CONC) .OR. EXCHR) GO TO 41
C
C     PRINT MOLAR CONCENTRATIONS AND REACTION CONVERSION RATES
      WRITE (6,136)
```

```
  136 FORMAT (1X,7HSPECIES,4X,13HCONCENTRATION,3X,13HMOLE FRACTION,3X,  OUTPP351
     * 27HNET SPECIES PRODUCTION RATE,5X,8HREACTION,3X,28HNET REACTION COUTPP352
     *ONVERSION RATE,3X,13HRATE CONSTANT)                                OUTPP353
      WRITE (6,137)                                                      OUTPP354
  137 FORMAT (12X,13H(MOLES/FT**3),25X,16H(MOLE/FT**3/SEC),11X,6HNUMBER,OUTPP355
     * 7X,22H(MOLE-FT**3/LB**2/SEC),7X,11H(CGS UNITS))                   OUTPP356
      DO 40 J=1,IR                                                       OUTPP357
   40 PRX(J) = XX(J)                                                     OUTPP358
      CONV = 1./62.43                                                    OUTPP359
      GO TO 42                                                           OUTPP360
C                                                                        OUTPP361
C     PRINT MOLAR CONCENTRATIONS AND ENERGY EXCHANGE RATES               OUTPP362
   41 WRITE (6,138)                                                      OUTPP363
  138 FORMAT (1X,7HSPECIES,4X,13HCONCENTRATION,3X,13HMOLE FRACTION,3X,  OUTPP364
     * 27HNET SPECIES PRODUCTION RATE,5X,8HREACTION,5X,20HNET ENERGY EXCOUTPP365
     *HANGE RATE,5X,13HRATE CONSTANT)                                    OUTPP366
      WRITE (6,139)                                                      OUTPP367
  139 FORMAT (12X,13H(MOLES/FT**3),25X,16H(MOLE/FT**3/SEC),11X,6HNUMBER,OUTPP368
     * 8X,21H(BTU-FT**3/LB**2/SEC),7X,11H(CGS UNITS))                    OUTPP369
C                                                                        OUTPP370
C     COMPUTE MOLAR CONCENTRATIONS                                       OUTPP371
   42 DO 43 I=1,LS                                                       OUTPP372
   43 PRC(I) = SIGMA(I)*RHOO                                             OUTPP373
C                                                                        OUTPP374
   44 DO 47 IJ=1,MAX                                                     OUTPP375
      IF (IJ .GT. LS .OR. IJ .GT. LR) GO TO 45                           OUTPP376
      FMOL = SIGMA(IJ)*MIXMW                                             OUTPP377
      WW = W(IJ)*62.43                                                   OUTPP378
      XXX = PRX(IJ)*CONV                                                 OUTPP379
      WRITE (6,140) (SPNM(I,IJ),I=1,2),PRC(IJ),FMOL,WW,IJ,XXX,RATE(IJ)   OUTPP380
  140 FORMAT (2X,2A4,2X,1PE12.5,4X,E12.5,11X,E12.5,16X,I2,14X,E12.5,11X,OUTPP381
     * E12.5)                                                            OUTPP382
      GO TO 47                                                           OUTPP383
   45 IF (IJ .GT. LS) GO TO 46                                           OUTPP384
      FMOL = SIGMA(IJ)*MIXMW                                             OUTPP385
      WW = W(IJ)*62.43                                                   OUTPP386
      WRITE (6,141) (SPNM(I,IJ),I=1,2),PRC(IJ),FMOL,WW                   OUTPP387
  141 FORMAT (2X,2A4,2X,1PE12.5,4X,E12.5,11X,E12.5)                      OUTPP388
      GO TO 47                                                           OUTPP389
   46 XXX = PRX(IJ)*CONV                                                 OUTPP390
      WRITE (6,142) IJ,XXX,RATE(IJ)                                      OUTPP391
  142 FORMAT (79X,I2,14X,1PE12.5,11X,E12.5)                              OUTPP392
   47 CONTINUE                                                           OUTPP393
      TXXH = TXXH*0.02883                                                OUTPP394
      WRITE (6,143) MIXMW,TXXH,CSUM                                      OUTPP395
  143 FORMAT ( /4X,20HMIXTURE MOLECULAR WEIGHT,F13.5,5X,26HTOTAL ENERGY OUTPP396
     *EXCHANGE RATE,1PE15.5,7X,17HMASS FRACTION SUM,0PF14.8)             OUTPP397
      WRITE (6,144)                                                      OUTPP398
  144 FORMAT (49X,21H(BTU-FT**3/LB**2/SEC))                              OUTPP399
      GO TO 78                                                           OUTPP400
```

```
C
   48 IF (UNITO .NE. SI) GO TO 63
C
C     CONVERT FROM INTERNAL (CGS) UNITS TO SI UNITS
      X = X*.01
      AREAA = AREA*.0001
      DOTM = MDOT*0.001
      PP = P*1.01325E+05
      VV = V*.01
      RHOO = RHO*1000.
      TENT = TENT*4184.0
C
      WRITE (6,145) TIME,AREAA,X,PP,NH,VV,AVH,RHOO,(SNAM(I,JCV),I=1,2),
     * T,DOTM,TENT,ERRN,MACH,KOUNT,GAMMA
  145 FORMAT (16X,4HTIME,1PE14.5,5H  SEC,10X,4HAREA,E14.5,7H  SQ M ,
     * 14X,14HAXIAL POSITION,E14.5,4H  M ///20X,15HFLOW PROPERTIES,45X,
     *    22HINTEGRATION INDICATORS//22X,8HPRESSURE,E22.5,30X,21HSTEPS FR
     *OM LAST PRINT,9X,I4/23X,8H(N/M**2)/22X,8HVELOCITY,E22.5,30X,17HAVE
     *RAGE STEP SIZE,0PE24.5/23X,7H(M/SEC)/22X,7HDENSITY,1PE23.5,30X,
     * 20HCONTROLLING VARIABLE,11X,2A4/23X,9H(KG/M**3)/22X,11HTEMPERATUR
     *E,E19.5/23X,7H(DEG K)/22X,14HMASS FLOW RATE,E16.5/23X,8H(KG/SEC)/
     * 22X,7HENTROPY,E23.5,30X,14HRELATIVE ERROR,0PE27.5/23X,16H(JOULE/K
     *G/DEG K)/22X,11HMACH NUMBER,1PE19.5,30X,20HPREDICTOR ITERATIONS,
     * 11X,I3//22X,5HGAMMA,E25.5)
C
      IF (I2 .EQ. 0) GO TO 49
      IS = I1 + 1
      DO 248 K=IS,IT
      KK = NEGL(K)
      ESP(1,K) = SPNM(1,KK)
  248 ESP(2,K) = SPNM(2,KK)
      WRITE (6,130) ((ESP(I,K),I=1,2),K=IS,IT)
C
   49 WRITE (6,131)
C
      CONV = 4.1840
      IF (CONC .OR. EXCHR) GO TO 51
C
C     PRINT MASS FRACTIONS AND REACTION CONVERSION RATES
      WRITE (6,132)
      WRITE (6,146)
  146 FORMAT (50X,15H(MOLE/M**3/SEC),12X,6HNUMBER,7X,21H(MOLE-M**3/KG**2
     */SEC),8X,11H(CGS UNITS))
      DO 50 J=1,IR
   50 PRX(J) = XX(J)
      CONV = 0.001
      GO TO 52
C
   51 IF (CONC .OR. (.NOT. EXCHR)) GO TO 54
C
```

```
C     PRINT MASS FRACTIONS AND ENERGY EXCHANGE RATES
      WRITE (6,134)
      WRITE (6,147)
  147 FORMAT (50X,15H(MOLE/M**3/SEC),12X,6HNUMBER,7X,22H(JOULE-M**3/KG**
     *2/SEC),7X,11H(CGS UNITS))
C
C     COMPUTE MASS FRACTIONS
   52 DO 53 I=1,LS
   53 PRC(I) = SIGMA(I)*MW(I)
      GO TO 59
C
   54 IF ((.NOT. CONC) .OR. EXCHR) GO TO 56
C
C     PRINT MOLAR CONCENTRATIONS AND REACTION CONVERSION RATES
      WRITE (6,136)
      WRITE (6,148)
  148 FORMAT (12X,12H(MOLES/M**3),26X,15H(MOLE/M**3/SEC),12X,6HNUMBER,7X
     *,21H(MOLE-M**3/KG**2/SEC),8X,11H(CGS UNITS))
      DO 55 J=1,LR
   55 PRX(J) = XX(J)
      CONV = 0.001
      GO TO 57
C
C     PRINT MOLAR CONCENTRATIONS AND ENERGY EXCHANGE RATES
   56 WRITE (6,138)
      WRITE (6,149)
  149 FORMAT (12X,12H(MOLES/M**3),26X,15H(MOLE/M**3/SEC),12X,6HNUMBER,7X
     *,22H(JOULE-M**3/KG**2/SEC),7X,11H(CGS UNITS))
C
C     COMPUTE MOLAR CONCENTRATIONS
   57 DO 58 I=1,LS
   58 PRC(I) = SIGMA(I)*RHOO
C
   59 DO 62 IJ=1,MAX
      IF (IJ .GT. LS .OR. IJ .GT. LR) GO TO 60
      FMOL = SIGMA(IJ)*MIXMW
      WW = W(IJ)*1000.
      XXX = PRX(IJ)*CONV
      WRITE (6,140) (SPNM(I,IJ),I=1,2),PRC(IJ),FMOL,WW,IJ,XXX,RATE(IJ)
      GO TO 62
   60 IF (IJ .GT. LS) GO TO 61
      FMOL = SIGMA(IJ)*MIXMW
      WW = W(IJ)*1000.
      WRITE (6,141) (SPNM(I,IJ),I=1,2),PRC(IJ),FMOL,WW
      GO TO 62
   61 XXX = PRX(IJ)*CONV
      WRITE (6,142) IJ,XXX,RATE(IJ)
   62 CONTINUE
      TXXH = TXXH*4.1840
      WRITE (6,143) MIXMW,TXXH,CSUM
```

```
      WRITE (6,150)
  150 FORMAT (48X,22H(JOULE-M**3/KG**2/SEC))
      GO TO 78
C
C     PRINT OUTPUT IN INTERNAL (CGS) UNITS
   63 WRITE (6,151) TIME,AREA,X,P,NH,V,AVH,RHO,(SNAM(I,JCV),I=1,2),
     * T,MDOT,TENT,ERRN,MACH,KOUNT,GAMMA
  151 FORMAT (16X,4HTIME,1PE14.5,5H  SEC,10X,4HAREA,E14.5,7H  SQ CM,
     * 14X,14HAXIAL POSITION,E14.5,4H  CM///20X,15HFLOW PROPERTIES,45X,2
     *2HINTEGRATION INDICATORS//22X,8HPRESSURE,0PF23.5,29X,21HSTEPS FROM
     * LAST PRINT,9X,I4/23X,5H(ATM)/22X,8HVELOCITY,F20.2,32X,17HAVERAGE
     *STEP SIZE,E24.5/23X,8H(CM/SEC)/22X,7HDENSITY,1PE23.5,30X,20HCONTRO
     *LLING VARIABLE,11X,2A4/23X,10H(GM/CM**3)/22X,11HTEMPERATURE,
     * 0PF17.2/23X,7H(DEG K)/22X,14HMASS FLOW RATE,1PE16.5/23X,8H(GM/SEC
     *)/22X,7HENTROPY,0PF23.4,30X,14HRELATIVE ERROR,E27.5/23X,14H(CAL/GM
     */DEG K)/22X,11HMACH NUMBER,F19.4,30X,20HPREDICTOR ITERATIONS,11X,
     * I3//22X,5HGAMMA,F25.4)
C
      IF (I2 .EQ. 0) GO TO 64
      IS = I1 + 1
      DO 263 K=IS,IT
      KK = NEGL(K)
      ESP(1,K) = SPNM(1,KK)
  263 ESP(2,K) = SPNM(2,KK)
      WRITE (6,130) ((ESP(I,K),I=1,2),K=IS,IT)
C
   64 WRITE (6,131)
C
      IF (CONC .OR. EXCHR) GO TO 66
C
C     PRINT MASS FRACTIONS AND REACTION CONVERSION RATES
      WRITE (6,132)
      WRITE (6,152)
  152 FORMAT (50X,16H(MOLE/CM**3/SEC),11X,6HNUMBER,7X,22H(MOLE-CM**3/GM*
     **2/SEC),7X,11H(CGS UNITS))
      DO 65 J=1,IR
   65 PRX(J) = XX(J)
      GO TO 67
C
   66 IF (CONC .OR. (.NOT. EXCHR)) GO TO 69
C
C     PRINT MASS FRACTIONS AND ENERGY EXCHANGE RATES
      WRITE (6,134)
      WRITE (6,153)
  153 FORMAT (50X,16H(MOLE/CM**3/SEC),11X,6HNUMBER,8X,21H(CAL-CM**3/GM**
     *2/SEC),7X,11H(CGS UNITS))
C
C     COMPUTE MASS FRACTIONS
   67 DO 68 I=1,IS
   68 PRC(I) = SIGMA(I)*MW(I)
```

```
      GO TO 74                                                          OUTPP551
C                                                                       OUTPP552
   69 IF ((.NOT. CONC) .OR. EXCHR) GO TO 71                             OUTPP553
C                                                                       OUTPP554
C     PRINT MOLAR CONCENTRATIONS AND REACTION CONVERSION RATES          OUTPP555
      WRITE (6,136)                                                     OUTPP556
      WRITE (6,154)                                                     OUTPP557
  154 FORMAT (12X,13H(MOLES/CM**3),25X,16H(MOLE/CM**3/SEC),11X,6HNUMBER,OUTPP558
     * 7X,22H(MOLE-CM**3/GM**2/SEC),7X,11H(CGS UNITS))                  OUTPP559
      DO 70 J=1,LR                                                      OUTPP560
   70 PRX(J) = XX(J)                                                    OUTPP561
      GO TO 72                                                          OUTPP562
C                                                                       OUTPP563
C     PRINT MOLAR CONCENTRATIONS AND ENERGY EXCHANGE RATES              OUTPP564
   71 WRITE (6,138)                                                     OUTPP565
      WRITE (6,155)                                                     OUTPP566
  155 FORMAT (12X,13H(MOLES/CM**3),25X,16H(MOLE/CM**3/SEC),11X,6HNUMBER,OUTPP567
     * 8X,21H(CAL-CM**3/GM**2/SEC),7X,11H(CGS UNITS))                   OUTPP568
C                                                                       OUTPP569
C     COMPUTE MOLAR CONCENTRATIONS                                      OUTPP570
   72 DO 73 I=1,LS                                                      OUTPP571
   73 PRC(I) = SIGMA(I)*RHO                                             OUTPP572
C                                                                       OUTPP573
   74 DO 77 IJ=1,MAX                                                    OUTPP574
      IF (IJ .GT. LS  .OR.  IJ .GT. LR) GO TO 75                        OUTPP575
      FMOL = SIGMA(IJ)*MIXMW                                            OUTPP576
      WRITE (6,140) (SPNM(I,IJ),I=1,2),PRC(IJ),FMOL,W(IJ),IJ,PRX(IJ),   OUTPP577
     * RATE(IJ)                                                         OUTPP578
      GO TO 77                                                          OUTPP579
   75 IF (IJ .GT. LS) GO TO 76                                          OUTPP580
      FMOL = SIGMA(IJ)*MIXMW                                            OUTPP581
      WRITE (6,141) (SPNM(I,IJ),I=1,2),PRC(IJ),FMOL,W(IJ)               OUTPP582
      GO TO 77                                                          OUTPP583
   76 WRITE (6,142) IJ,PRX(IJ),RATE(IJ)                                 OUTPP584
   77 CONTINUE                                                          OUTPP585
      WRITE (6,143) MIXMW,TXXH,CSUM                                     OUTPP586
      WRITE (6,156)                                                     OUTPP587
  156 FORMAT (49X,21H(CAL-CM**3/GM**2/SEC))                             OUTPP588
C                                                                       OUTPP589
   78 WRITE (6,157)                                                     OUTPP590
  157 FORMAT (//2(4X,8HVARIABLE,5X,10HDERIVATIVE,6X,9HINCREMENT,6X,14HREOUTPP591
     *LATIVE ERROR,4X))                                                 OUTPP592
      L = LSP3/2                                                        OUTPP593
C                                                                       OUTPP594
      LP = L                                                            OUTPP595
      DO 79 I=1,L                                                       OUTPP596
      LP = LP + 1                                                       OUTPP597
      WRITE (6,158) (SNAM(K,I),K=1,2),F(I),RK(I),E(I),                  OUTPP598
     * (SNAM(K,LP),K=1,2),F(LP),RK(LP),E(LP)                            OUTPP599
  158 FORMAT (2(4X,2A4,2(4X,E12.5),5X,F12.5,5X))                        OUTPP600
```

```
   79 CONTINUE                                                          OUTPP601
C                                                                       OUTPP602
      IF (2*L .EQ. LSP3) GO TO 80                                       OUTPP603
      WRITE (6,159) (SNAM(K,LSP3),K=1,2),F(LSP3),RK(LSP3),E(LSP3)       OUTPP604
  159 FORMAT (70X,2A4,2(4X,E12.5),5X,E12.5)                             OUTPP605
C                                                                       OUTPP606
   80 IF (.NOT. DBUGO) GO TO 82                                         OUTPP607
C     DEBUG OUTPUT (INTERNAL UNITS)                                     OUTPP608
      IF (VERSA .NE. AREAV) GO TO 83                                    OUTPP609
      WRITE (6,160) AA,BB,DA,D2A,DTERM                                  OUTPP610
  160 FORMAT (1H1,12X,2HAA,18X,2HBB,13X,15HD(AREA)/D(IVAR),8X,17HD2(AREAOUTPP611
     *)/D(IVAR)2,8X,27H(1/AREA)*D(AREA)/D(IVAR)-AA/8X,E12.5,8X,E12.5,9X,OUTPP612
     * E12.5,12X,E12.5,18X,E12.5)                                       OUTPP613
      GO TO 84                                                          OUTPP614
   83 WRITE (6,164) AA,BB,DA,D2A,DTERM                                  OUTPP615
  164 FORMAT (1H1,12X,2HAA,18X,2HBB,11X,19HD(PRESSURE)/D(IVAR),4X,21HD2(OUTPP616
     *PRESSURE)/D(IVAR)2,4X,32H(1/PRESSURE)*D(PRESSURE)/D(IVAR)/8X,E12.5OUTPP617
     *,8X,E12.5,9X,E12.5,12X,E12.5,18X,E12.5)                           OUTPP618
   84 WRITE (6,161) (I,I=1,LR)                                          OUTPP619
  161 FORMAT (/37X,58HOMEGA(I,J)   RATE OF PRODUCTION OF SPECIES I BY REOUTPP620
     *ACTION J//1X,7HSPECIES,55X,8HREACTION/(18X,I2,7I15))              OUTPP621
      DO 81 I=1,LS                                                      OUTPP622
      WRITE (6,162) (SPNM(K,I),K=1,2),(OMEGA(I,J),J=1,LR)               OUTPP623
  162 FORMAT (/2X,2A4,8E15.5/(10X,8E15.5))                              OUTPP624
   81 CONTINUE                                                          OUTPP625
C                                                                       OUTPP626
   82 IF (ABS(1.-CSUM) .LE. .001) RETURN                                OUTPP627
      WRITE (6,163)                                                     OUTPP628
  163 FORMAT (7H0(OUTP),5X,19HINVALID COMPOSITION)                      OUTPP629
      NEXT = .TRUE.                                                     OUTPP630
      RETURN                                                            OUTPP631
C                                                                       OUTPP632
C    OPTION ABBREVIATED OUTPUT DIVERTED TO POSTPROCESSOR                OUTPP633
C                                                                       OUTPP634
   90 WRITE(3)KEY,TIME,AREA,X,P,V,RHO,T,MDOT,TENT,MACH,GAMMA,MIXMW,     OUTPP635
     'SIGMA,DUM1                                                        OUTPP636
      WRITE(6,165)TIME,AREA,X,NH,AVH                                    OUTPP637
  165 FORMAT(7H0TIME= ,1PE13.6,15H SEC      AREA= ,E13.6,11H SQ CM     , OUTPP638
     '    3HX= ,E13.6,19H CM      NO. STEPS= ,I4,21H     AVG. STEP SIZE= ,OUTPP639
     'E13,6)                                                            OUTPP640
      CALL CLOCK(TWALL,TPROC,MONTH,IDAY)                                OUTPP641
      IF(TPROC.LT.(36.*TIMLMT))GO TO 82                                 OUTPP642
      WRITE(6,166)                                                      OUTPP643
  166 FORMAT(40H0(OUTP)      STOP - EXCESS PROCESSOR TIME)              OUTPP644
      STOP                                                              OUTPP645
      END                                                               OUTPP646
```

```
CPADD              ADDITION OF MESH POINTS AT JET EDGE
      SUBROUTINE PADD(YM,NPD,REG)
      LOGICAL  FIRST
      LOGICAL MCHANG
      INTEGER REG
      COMMON /UMESH/ MCHANG,CK,DY1,NMSH      , CXPC,CXTP,NRED
      COMMON/ INPJET/ DUMI(12),  NJ  , DUMK(208)
      COMMON /PROPJT/ DUMP(9),REFL,DUMP1(1403)
      COMMON /CTRL  / DUMSX(9),C6,DUMSX1(809)
      DIMENSION YM(1)
      DATA FIRST /T/
C*
C***  REG=1-- USE CONSTANT MESH SIZE  (POTENTIAL CORE)
C**   REG=2,3-- USE MESH DEFINED BY /UMESH/
C*
      IF(.NOT. MCHANG) GO TO 10
    1 GO TO (10,20,20),REG
   10 YK=YM(NPD)
      DY     = .01*YK
   11 NPD=NPD+1
      IF(.NOT.FIRST) DY=.1*CXPC*REFL/C6
      FIRST = .FALSE.
      YM(NPD)=YK+DY
      GO TO 100
C*
   20 DYF=YM(2)-YM(1)
      NEW=NPD+1
      YM(NEW)=DYF/(CK-1.)*(CK**FLOAT(NPD)-1.)
      NPD=NEW
C*
  100 RETURN
      END
```

```
CPBOLIC       COMPUTES SOLUTION OF PARABOLIC PDE
*
      SUBROUTINE PBOLIC (K)
*
C -J- IS THE CROSS STREAM INDEX
C -K- IS THE INDEX FOR DEPENDENT VARIABLES
C THETA IS THE CRANK-NICOLSON FACTOR
C THETA=0-EXPLICIT, THETA=1-IMPLICIT, THETA=1/2,CRANK-NICOLSON
*
      COMMON /CTHETA/ THETA,II
      COMMON /CENDS/ JSTART,JEND
      COMMON /CPBOLI/ G(50,1),ALPHA(1,1),BETA(50,1),GAMM(50,1),
     1  DELTA(50,1)
      COMMON /CTRIDI/ COEFL(50),COEFC(50),COEFR(50),RHS(50)
      COMMON /CBNDRY/ DX,DXPRN,JT
      COMMON /CINIT/ XX,DPSI(1)
C
      NJM1=JEND-1
  100 JSP1=JSTART+1
      THETA1=1.0-THETA
C*  SETUP RHS AND THEN UP-DATE BETA,GAMM,DELTA FOR NEW X-STEP
      AAA=ALPHA(II,K)*DX/(DPSI(JT)*DPSI(JT))
      AAAA=AAA*THETA1*.50
      DO 200 J=JSP1,NJM1
      J1=J
      COEFL(J1)=AAAA*(BETA(J,K)+BETA(J-1,K))
      COEFR(J1)=AAAA*(BETA(J,K)+BETA(J+1,K))
  200 RHS(J1)=COEFL(J1)*G(J-1,K)+(1.-COEFL(J1)-COEFR(J1)+(THETA1*DX*
     1 DELTA(J,K)))*G(J,K)+COEFR(J1)*G(J+1,K)+THETA1*GAMM(J,K)*DX
C*  UPDATE COEFFICIENTS --BETA,GAMM,DELTA--
  260 CALL COEFF(K)
      AAA=ALPHA(II,K)*DX/(DPSI(JT)*DPSI(JT))
      AAAA=THETA*AAA*.5
C*  RHS USES GAMM AT BOTH OLD AND NEW STATIONS
      DO 300 J=JSP1,NJM1
      J1=J
      RHS(J1)=RHS(J1)+THETA*DX*GAMM(J,K)
      COEFR(J1)=-AAAA*(BETA(J,K)+BETA(J+1,K))
      COEFL(J1)=-AAAA*(BETA(J,K)+BETA(J-1,K))
  300 COEFC(J1)=1.-COEFL(J1)-COEFR(J1)-THETA*DX*DELTA(J,K)
C*  AXIS OF SYMMETRY B.C.
  310 J1=JSP1
      COEFR(J1)=COEFR(J1)-.33333333*COEFL(J1)
      COEFC(J1)=COEFC(J1)+1.33333333*COEFL(J1)
C*  OUTER (FREE-STREAM) B.C.
  360 RHS(NJM1)=RHS(NJM1)-COEFR(NJM1)*G(JEND,K)
      CALL TRIDIA(JSP1,NJM1)
  390 DO 400 J=JSP1,NJM1
      J1=J
      G(J1,K)=RHS(J)
```

```
      IF (G(J1,K).LT.(1.E-8)) G(J1,K)=0.                              PBOLIC51
  400 CONTINUE                                                        PBOLIC52
      G(1,K)=(4.*G(2,K)-G(3,K))*.33333333                             PBOLIC53
      IF (G(1,K).LT.(1.E-8)) G(1,K)=0.                                PBOLIC54
      RETURN                                                          PBOLIC55
      END                                                             PBOLIC56
```

```
CPARDD      COMPUTE ALL MIXED PARTIAL DERIVATIVES
      SUBROUTINE PARD
C
C     COMPUTE ALL MIXED PARTIAL DERIVATIVES
C
      LOGICAL TCON,RHOCON
C
      REAL LKEQ,MM,N,M,MIXMW,M2
C
      DIMENSION PXXRHO(30),PXXT(30),PXXSIG(30,25),PGSIG(25),PM2SIG(25),
     * PS1SIG(25),PS2SIG(25),PAASIG(25),PBBSIG(25)
C
      COMMON/OPTS/VERSI,TIMEV,VERSA,AREAV,ELIM,TCON,RHOCON,IPRCOD
      COMMON/COND/DVAR,AREA,MDOT,P,IVAR,V,RHO,T,SIGMA(25),LS,LSP3,NEXT
      COMMON/SPEC/SNAM(2,30),MW(25),W(25),STOIC(25,30),OMEGA(25,30)
      COMMON/REAC/LSR(4,30),XX(30),RATE(30),LKEQ(30),DLKEQ(30),MM(30),LR
      COMMON/RRAT/A(30),N(30),EACT(30),B(30),M(25,30),ALLM1
      COMMON/GHSC/GRT(25),HRT(25),SR(25),CPR(25),DCPR(25)
      COMMON/NECC/RR,MIXMW,M2,GAMMA,TCPR,R
      COMMON/SABS/S1,AA,BB,S2,DA,D2A,DTERM,IRHO
      COMMON/DERN/F(28),ALPHA(28),BETA(28,28)
C
      DO 1 I=1,LSP3
      DO 1 K=1,LSP3
    1 BETA(I,K) = 0.
C
      DO 2 J=1,LR
      DO 2 I=1,LS
    2 PXXSIG(J,I) = 0.
C
C     XX(J) WRT RHO,T,SIGMA(I)
      DO 9 J=1,LR
      N1 = LSR(1,J)
      N2 = LSR(2,J)
      N3 = LSR(3,J)
      N4 = LSR(4,J)
      EXP1 = RATE(J)
      EXP2 = 1.
      IF (LKEQ(J) .GT. 0.) GO TO 3
      EXP2 = EXP(-LKEQ(J)/2.)
      GO TO 4
    3 EXP1 = EXP(ALOG(RATE(J)) - LKEQ(J))
C
    4 IF (N1 .GT. 0) GO TO 6
      IF (N1 .LT. 0) GO TO 5
      PXXRHO(J) = -MM(J)*EXP2*SIGMA(N3)*EXP1*SIGMA(N4)*EXP2
      PXXT(J) = MM(J)*EXP1*RHO*SIGMA(N3)*EXP2*SIGMA(N4)*EXP2*DLKEQ(J)
      PXXSIG(J,N2) = RATE(J)*MM(J) + XX(J)/MM(J)*M(N2,J)
      PXXSIG(J,N3) = -MM(J)*EXP1*RHO*EXP2*SIGMA(N4)*EXP2 + XX(J)/MM(J)*
     * M(N3,J)
```

```
      PXXSIG(J,N4) = -MM(J)*EXP1*RHO*EXP2*SIGMA(N3)*EXP2
      IF(N4.EQ.N3)PXXSIG(J,N4)=2.*PXXSIG(J,N4)
      PXXSIG(J,N4)=PXXSIG(J,N4)+XX(J)/MM(J)*M(N4,J)
      GO TO 9
    5 PXXRHO(J) = -RATE(J)/RHO*SIGMA(N2)/RHO
      PXXT(J) = EXP1*SIGMA(N3)*EXP2*SIGMA(N4)*EXP2*DLKEQ(J)
      PXXSIG(J,N2) = RATE(J)/RHO
      PXXSIG(J,N3) = -EXP2*(EXP1*SIGMA(N4))*EXP2
      PXXSIG(J,N4) = -EXP2*(EXP1*SIGMA(N3))*EXP2
      IF(N4.EQ.N3)PXXSIG(J,N4)=2.*PXXSIG(J,N4)
      GO TO 9
C
    6 IF (N4 .GT. 0) GO TO 8
      IF (N4 .LT. 0) GO TO 7
      PXXRHO(J) = MM(J)*SIGMA(N1)*RATE(J)*SIGMA(N2)
      PXXT(J) = EXP2*MM(J)*EXP1*SIGMA(N3)*EXP2*DLKEQ(J)
      PXXSIG(J,N1) = MM(J)*RHO*RATE(J)*SIGMA(N2) + XX(J)/MM(J)*M(N1,J)
      PXXSIG(J,N2) = MM(J)*RHO*RATE(J)*SIGMA(N1)
      IF(N2.EQ.N1)PXXSIG(J,N2)=2.*PXXSIG(J,N2)
      PXXSIG(J,N2)=PXXSIG(J,N2)+XX(J)/MM(J)*M(N2,J)
      PXXSIG(J,N3) = -EXP1*(EXP2*MM(J))*EXP2 + XX(J)/MM(J)*M(N3,J)
      GO TO 9
    7 PXXRHO(J) = EXP1*SIGMA(N3)*EXP2/RHO*EXP2/RHO
      PXXT(J) = EXP1*SIGMA(N3)*EXP2/RHO*EXP2*DLKEQ(J)
      PXXSIG(J,N1) = RATE(J)*SIGMA(N2)
      PXXSIG(J,N2) = RATE(J)*SIGMA(N1)
      IF(N2.EQ.N1)PXXSIG(J,N2)=2.*PXXSIG(J,N2)
      PXXSIG(J,N3) = -EXP2*EXP1*(EXP2/RHO)
      GO TO 9
C
    8 PXXRHO(J) = 0.
      PXXT(J) = EXP1*SIGMA(N3)*EXP2*SIGMA(N4)*EXP2*DLKEQ(J)
      PXXSIG(J,N1) = RATE(J)*SIGMA(N2)
      PXXSIG(J,N2) = RATE(J)*SIGMA(N1)
      IF(N2.EQ.N1)PXXSIG(J,N2)=2.*PXXSIG(J,N2)
      PXXSIG(J,N3) = -EXP2*(EXP1*SIGMA(N4))*EXP2
      PXXSIG(J,N4) = -EXP2*(EXP1*SIGMA(N3))*EXP2
      IF(N4.EQ.N3)PXXSIG(J,N4)=2.*PXXSIG(J,N4)
C
    9 PXXT(J) = PXXT(J) + XX(J)*(N(J) + B(J)/T)/T
C
      GTGM1 = GAMMA*(GAMMA - 1.)
      PGAMT = 0.
C   GAMMA WRT SIGMA(I)  AND  MACH NUMBER SQUARED WRT SIGMA(I)
      DO 10 I=1,LS
      PGSIG(I) = GTGM1*(MIXMW - CPR(I)/TCPR)
      PM2SIG(I) = -M2*(MIXMW + PGSIG(I)/GAMMA)
   10 PGAMT = PGAMT + SIGMA(I)*DCPR(I)
C
C   GAMMA WRT T
```

```
      PGAMT = -GTGM1/TCPR*PGAMT
C
C     MACH NUMBER SQUARED WRT V
      PM2V = 2.*V*MIXMW/(GAMMA*R*T)
C     MACH NUMBER SQUARED WRT T
      PM2T = -M2*(1./T + PGAMT/GAMMA)
C
      TERM = RHO
      IF (VERST .EQ. TIMEV) GO TO 12
      TERM = RHO/V
C     DSIGMA/DIVAR WRT V
      DO 11 II=4,LSP3
   11 BETA(II,1) = -F(II)/V
C
C     DSIGMA/DIVAR WRT RHO AND DSIGMA/DIVAR WRT T
   12 DO 14 II=4,LSP3
      I = II - 3
      DO 13 J=1,IR
      BETA(II,2) = BETA(II,2) + STOIC(I,J)*PXXRHO(J)
   13 BETA(II,3) = BETA(II,3) + STOIC(I,J)*PXXT(J)
      BETA(II,2) = F(II)/RHO + TERM*BETA(II,2)
   14 BETA(II,3) = TERM*BETA(II,3)
C
C     DSIGMA(I)/DIVAR WRT SIGMA(K)
      DO 16 II=4,LSP3
      I = II - 3
      DO 16 KK=4,LSP3
      K = KK - 3
      DO 15 J=1,IR
   15 BETA(II,KK) = BETA(II,KK) + STOIC(I,J)*PXXSIG(J,K)
   16 BETA(II,KK) = TERM*BETA(II,KK)
C
C     S1 WRT V,RHO,T,SIGMA(I)  AND  S2 WRT V,RHO,T,SIGMA(I)
      PS1V = 0.
      PS1RHO = 0.
      PS1T = 0.
      PS2V = 0.
      PS2RHO = 0.
      PS2T = 0.
      DO 18 II=4,LSP3
      I = II - 3
      PS1V = PS1V + BETA(II,1)
      PS1RHO = PS1RHO + BETA(II,2)
      PS1T = PS1T + BETA(II,3)
      PS2V = PS2V + HRT(I)*BETA(II,1)
      PS2RHO = PS2RHO + HRT(I)*BETA(II,2)
      PS2T = PS2T + HRT(I)*BETA(II,3) + CPR(I)*F(II)/T
      PS1SIG(I) = 0.
      PS2SIG(I) = 0.
      DO 17 KK=4,LSP3
```

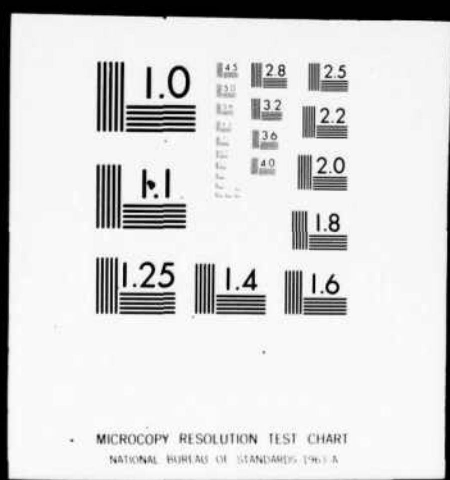
AD-A045 627 GENERAL ELECTRIC CO CINCINNATI OHIO AIRCRAFT ENGINE GROUP F/G 21/5
DEVELOPMENT OF EMISSIONS MEASUREMENT TECHNIQUES FOR AFTERBURNIN--ETC(U)
OCT 75 W C COLLEY, D R FERGUSON, M A SMITH F33615-73-C-2047
UNCLASSIFIED R75AEG459 AFAPL-TR-75-52-SUPPL-2 NL

5 OF 5
ADA045 627

END
DATE
FILMED
11-77
DDC

5 OF 5

ADA045 627



```
      K = KK - 3
      PS1SIG(I) = PS1SIG(I) + BETA(KK,II)
   17 PS2SIG(I) = PS2SIG(I) + HRT(K)*BETA(KK,II)
      PS1SIG(I) = MIXMW*(PS1SIG(I) - S1)
   18 PS2SIG(I) = MIXMW*(PS2SIG(I) - S2)
      PS1V = MIXMW*PS1V
      PS1RHO = MIXMW*PS1RHO
      PS1T = MIXMW*PS1T
      PS2V = MIXMW*PS2V
      PS2RHO = MIXMW*PS2RHO
      PS2T = MIXMW*PS2T - S2/T
C
      GM1DG = (GAMMA - 1.)/GAMMA
      S2OG2 = S2/(GAMMA*GAMMA)
C     BB WRT V
      PBBV = GM1DG*PS2V
C     BB WRT RHO
      PBBRHO = GM1DG*PS2RHO
C     BB WRT T
      PBBT = GM1DG*PS2T + S2OG2*PGAMT
C
C     AA WRT V
      PAAV = PS1V - PBBV
C     AA WRT RHO
      PAARHO = PS1RHO - PBBRHO
C     AA WRT T
      PAAT = PS1T - PBBT
C
C     BB WRT SIGMA(I)  AND  AA WRT SIGMA(I)
      DO 19 I=1,IS
      PBBSIG(I) = GM1DG*PS2SIG(I) + S2OG2*PGSIG(I)
   19 PAASIG(I) = PS1SIG(I) - PBBSIG(I)
C
      IF (VERSA .NE. AREAV) GO TO 24
C   ASSIGNED AREA EQUATIONS
      T1 = 1./(M2 - 1.)
      GAM1 = GAMMA - 1.
C
C     DV/DIVAR WRT V
      BETA(1,1) = T1*(DTERM - F(1)*PM2V - V*PAAV)
C     DV/DIVAR WRT RHO
      BETA(1,2) = -V*T1*PAARHO
C     DV/DIVAR WRT T
      BETA(1,3) = -T1*(V*PAAT + F(1)*PM2T)
C     DV/DIVAR WRT SIGMA(I)
      DO 20 II=4,LSP3
      I = II - 3
   20 BETA(1,II) = -T1*(V*PAASIG(I) + F(1)*PM2SIG(I))
C
      IF (RHOCON) GO TO 22
```

```
C     DRHO/DIVAR WRT V
      BETA(2,1) = RHO*T1*(PAAV + T1*DTERM*PM2V)
C     DRHO/DIVAR WRT RHO
      BETA(2,2) = RHO*T1*PAARHO + F(2)/RHO
C     DRHO/DIVAR WRT T
      BETA(2,3) = RHO*T1*(PAAT + T1*DTERM*PM2T)
C     DRHO/DIVAR WRT SIGMA(I)
      DO 21 II=4,LSP3
      I = II - 3
   21 BETA(2,II) = RHO*T1*(PAASIG(I) + T1*DTERM*PM2SIG(I))
C
   22 IF (TCON) RETURN
C     DT/DIVAR WRT V
      BETA(3,1) = T*(GAM1*T1*(M2*PAAV + T1*DTERM*PM2V) - PBBV)
C     DT/DIVAR WRT RHO
      BETA(3,2) = T*(GAM1*M2*T1*PAARHO - PBBRHO)
C     DT/DIVAR WRT T
      BETA(3,3) = T*(T1*(GAM1*(M2*PAAT + T1*DTERM*PM2T) - M2*DTERM*PGAMT
     * ) - PBBT) + F(3)/T
      IF(IRHO.EQ.2)BETA(3,3)=BETA(3,3)+T*GAM1*PAAT+T*AA*PGAMT
C     DT/DIVAR WRT SIGMA(I)
      DO 23 II=4,LSP3
      I = II - 3
      BETA(3,II)=T*(T1*(GAM1*(M2*PAASIG(I)+T1*DTERM*PM2SIG(I))-M2*
     * DTERM*PGSIG(I)) - PBBSIG(I))
      IF(IRHO.EQ.2)BETA(3,II)=BETA(3,II)+GAM1*T*PAASIG(I)+T*AA*PGSIG(I)
   23 CONTINUE
C
      RETURN
C   ASSIGNED PRESSURE EQUATIONS
   24 T1 = 1./(GAMMA*GAMMA)
C
C     DV/DIVAR WRT V
      IF (V .NE. 0.) BETA(1,1) = -F(1)/V
C     DV/DIVAR WRT RHO
      BETA(1,2) = -F(1)/RHO
C     DV/DIVAR WRT T
      BETA(1,3) = 0.
C     DV/DIVAR WRT SIGMA(I)
      DO 25 II=4,LSP3
   25 BETA(1,II) = 0.
C
      IF (RHOCON) GO TO 27
C     DRHO/DIVAR WRT V
      BETA(2,1) = -RHO*PAAV
C     DRHO/DIVAR WRT RHO
      BETA(2,2) = F(2)/RHO - RHO*PAARHO
C     DRHO/DIVAR WRT T
      BETA(2,3) = -RHO*(PAAT + T1*DTERM*PGAMT)
C     DRHO/DIVAR WRT SIGMA(I)
```

```
      DO 26 II=4,LSP3                                                   PARDD251
      I = II - 3                                                        PARDD252
   26 BETA(2,II) = -RHO*(PAASIG(I) + T1*DTERM*PGSIG(I))                PARDD253
C                                                                       PARDD254
   27 IF (TCON) RETURN                                                  PARDD255
C     DT/DIVAR WRT V                                                    PARDD256
      BETA(3,1) = -T*PBBV                                               PARDD257
C     DT/DIVAR WRT RHO                                                  PARDD258
      BETA(3,2) = -T*PBBRHO                                             PARDD259
C     DT/DIVAR WRT T                                                    PARDD260
      BETA(3,3) = BB - T*(PBBT - T1*DTERM*PGAMT) + F(3)/T               PARDD261
C     DT/DIVAR WRT SIGMA(I)                                             PARDD262
      DO 28 II=4,LSP3                                                   PARDD263
      I = II - 3                                                        PARDD264
   28 BETA(3,II) = -T*(PBBSIG(I) - T1*DTERM*PGSIG(I))                  PARDD265
C                                                                       PARDD266
      RETURN                                                            PARDD267
      END                                                               PARDD268
```

```
CPERRR      PREDICTS ERROR TO BE EXPECTED
      FUNCTION PERR (H)
C
C     THIS ROUTINE PREDICTS THE ERROR WHICH CAN BE EXPECTED FROM A
C     STEP OF SIZE H
C
      DOUBLE PRECISION SUM
C
      DIMENSION Y(28),RK(28),E(28)
C
      COMMON/COND/DUM(5),YN(28),LS,LSP3,NEXT
      COMMON/SINT/HMIN,HNM1,HN,HNP1,HMAX,NH,AVH,EMAX,ERRN,JCV,KOUNT,ERRP
      COMMON/DERN/F(28),ALPHA(28),BETA(28,28)
      COMMON/PQRF/PK(28),QK(28),DUM1(56)
      DATA JC/0/
      C = H/(2.*H + HN)
      C0 = H/2.
      C1 = H/HN
      C2 = H + HN
      PERR = 0.
C
      DO 2 I=1,LSP3
      RK(I) = C0*(QK(I)/HN + F(I))
    2 E(I) = C*(C1*QK(I) + C2*(F(I) + ALPHA(I)*H))
C
      C0 = C*C2
      DO 4 I=1,LSP3
      SUM = 0.D0
      DO 3 J=1,LSP3
      IF (J .EQ. I) GO TO 3
      SUM = SUM + BETA(I,J)*RK(J)
    3 CONTINUE
      RK(I) = (E(I) + C0*SUM)/(1. - C0*BETA(I,I))
    4 CONTINUE
C
      DO 6 I=1,LSP3
      SUM = 0.D0
      DO 5 J=1,LSP3
    5 SUM = SUM + BETA(I,J)*RK(J)
      RK(I) = F(I) + C0*SUM
    6 Y(I) = YN(I) + RK(I)
C
      PERR = ERRORC(Y,RK,E,JC,H)
C
      RETURN
      END
```

```
CPRAT         IMPACT/STATIC PRESSURE RATIO   PERFECT GAS
C
      FUNCTION PRAT(MACH,GAMMA)
      REAL MACH,MSQ
C
      G=GAMMA
      GP1=G+1.0
      GM1=G-1.0
      GEXP=G/GM1
C
      MSQ=MACH**2
      IF(MSQ.GT.1.0)GO TO 10
C
      PRAT=(1.0+0.5*GM1*MSQ)**GEXP
      RETURN
C
   10 DENOM=(2.0*G/GP1*MSQ-GM1/GP1)**(GEXP/G)
      PRAT=(0.5*GP1*MSQ)**GEXP/DENOM
      RETURN
      END
```

```
CPREDD       PERFORM ALL NECESSARY PRE-DERIVATIVE CALCULATIONS
      SUBROUTINE PRED
C
C     PERFORM ALL NECESSARY PRE-DERIVATIVE CALCULATIONS
C
      DOUBLE PRECISION DP1,DP2
C
      LOGICAL ALLM1,TCON,NEXT
C
      REAL IVAR,MDOT,LKEQ,MM,N,M,MIXMW,M2
C
      COMMON/OPTS/VERSI,TIMEV,VERSA,AREAV,FLIM,TCON,RHOCON,IPRCOD
      COMMON/COND/DVAR,AREA,MDOT,P,IVAR,V,RHO,T,SIGMA(25),LS,LSP3,NEXT
      COMMON/SPEC/SNAM(2,30),MW(25),W(25),STOIC(25,30),OMEGA(25,30)
      COMMON/REAC/LSR(4,30),XX(30),RATE(30),LKEQ(30),DLKEQ(30),MM(30),LR
      COMMON/RRAT/A(30),N(30),EACT(30),B(30),M(25,30),ALLM1
      COMMON/GHSC/GRT(25),HRT(25),SR(25),CPR(25),DCPR(25)
      COMMON/NECC/RR,MIXMW,M2,GAMMA,TCPR,R
      COMMON/SABS/S1,AA,BB,S2,DA,D2A,DTERM,IRHO
C
      IF (TCON) GO TO 5
      GO TO 1
C
      ENTRY PRED1
      MWARN = 0
C     THERMODYNAMIC PROPERTIES
    1 CALL THRM (T,1.)
C
      ALOGRT = ALOG(RR*T)
      DO 4 J=1,LR
C
C     REACTION RATE CONSTANT
      RATE(J) = A(J)*T**N(J)*EXP(-B(J)/T)
C
C     LN KEQ  AND  D(LN KEQ)/DT
      N1 = LSR(1,J)
      N2 = LSR(2,J)
      N3 = LSR(3,J)
      N4 = LSR(4,J)
      DELG = GRT(N3) - GRT(N2)
      DELH = HRT(N3) - HRT(N2)
      IF (N1 .GT. 0) GO TO 2
      DELG = DELG + GRT(N4)
      DELH = DELH + HRT(N4)
      LKEQ(J) = -DELG - ALOGRT
      DLKEQ(J) = (DELH - 1.)/T
      GO TO 4
    2 IF (N4 .GT. 0) GO TO 3
      DELG = DELG - GRT(N1)
      DELH = DELH - HRT(N1)
```

```
      LKEQ(J) = -DELG + ALOGRT
      DLKEQ(J) = (DELH + 1.)/T
      GO TO 4
    3 DELG = DELG + GRT(N4) - GRT(N1)
      DELH = DELH + HRT(N4) - HRT(N1)
      LKEQ(J) = -DELG
      DLKEQ(J) = DELH/T
    4 CONTINUE
C
C     MIXTURE MOLFCULAR WEIGHT
    5 SSUM = 0.
      DO 6 I=1,LS
    6 SSUM = SSUM + SIGMA(I)
      MIXMW = 1./SSUM
C
C     ASSIGNED VARIABLE
      IF (IPRCOD .GT. 2) GO TO 7
      X = IVAR
      IF (VERSI .EQ. TIMEV) X = DVAR
      CALL CINP (X,AVAR,DA,D2A)
      GO TO 8
    7 TIME = DVAR
      IF (VERSI .EQ. TIMEV) TIME = IVAR
      CALL CINP (TIME,AVAR,DA,D2A)
C
C     CALCULATED VARIABLE
    8 IF (VERSA .EQ. AREAV) GO TO 9
      P = AVAR
      IF (V .NE. 0.) AREA = MDOT/(RHO*V)
      GO TO 10
    9 AREA = AVAR
      P = RHO*RR*T/MIXMW
C     MASS FLOW RATE
      MDOT = RHO*AREA*V
C
   10 DO 20 J=1,IR
      N1 = LSR(1,J)
      N2 = LSR(2,J)
      N3 = LSR(3,J)
      N4 = LSR(4,J)
C
C     THIRD BODY FACTOR
      MM(J) = 0.
      IF (N1 .NE. 0   .AND.  N4 .NE. 0) GO TO 13
      IF (ALLM1) GO TO 12
      DO 11 I=1,LS
   11 MM(J) = MM(J) + M(I,J)*SIGMA(I)
      GO TO 13
   12 MM(J) = SSUM
C
```

```
   13 EXP1 = RATE(J)
      EXP2 = 1.
      IF (LKEQ(J) .GT. 0.) GO TO 14
      EXP2 = EXP(-LKEQ(J)/2.)
      GO TO 15
   14 EXP1 = EXP(ALOG(RATE(J)) - LKEQ(J))
C
C     NET REACTION CONVERSION RATE
   15 IF (N1 .GT. 0) GO TO 17
      IF (N1 .LT. 0) GO TO 16
      DP1 = RATE(J)*SIGMA(N2)
      DP2 = RHO*EXP1*SIGMA(N3)*EXP2*SIGMA(N4)*EXP2
      XX(J) = MM(J)*(DP1 - DP2)
      GO TO 20
   16 DP1 = RATE(J)*SIGMA(N2)/RHO
      DP2 = EXP1*SIGMA(N3)*EXP2*SIGMA(N4)*EXP2
      XX(J) = DP1 - DP2
      GO TO 20
   17 IF (N4 .GT. 0) GO TO 19
      IF (N4 .LT. 0) GO TO 18
      DP1 = RHO*SIGMA(N1)*RATE(J)*SIGMA(N2)
      DP2 = EXP1*EXP2*SIGMA(N3)*EXP2
      XX(J) = MM(J)*(DP1 - DP2)
      GO TO 20
   18 DP1 = SIGMA(N1)*RATE(J)*SIGMA(N2)
      DP2 = EXP1*SIGMA(N3)*EXP2/RHO*EXP2
      XX(J) = DP1 - DP2
      GO TO 20
   19 DP1 = SIGMA(N1)*RATE(J)*SIGMA(N2)
      DP2 = EXP2*SIGMA(N3)*EXP1*SIGMA(N4)*EXP2
      XX(J) = DP1 - DP2
   20 CONTINUE
C
      RHO2 = RHO*RHO
      TCPR = 0.
      DO 22 I=1,IS
C
C     TOTAL CP/R
      TCPR = TCPR + CPR(I)*SIGMA(I)
C
C     NET SPECIES PRODUCTION RATE
      W(I) = 0.
      DO 21 J=1,LR
      OMEGA(I,J) = RHO2*STOIC(I,J)*XX(J)
   21 W(I) = W(I) + OMEGA(I,J)
   22 CONTINUE
C
C     GAMMA (FROZEN)
      GAMMA = TCPR/(TCPR - 1./MIXMW)
C
```

```
C     MACH NUMBER SQUARED                                               PREDD151
      M2 = V/R*V/T*MIXMW/GAMMA                                          PREDD152
C                                                                       PREDD153
      IF (VERSA .NE. AREAV  .OR.  (M2 .LT. 0.9025 .OR. M2 .GT. 1.1025)) PREDD154
     * GO TO 23                                                         PREDD155
      WMACH = SQRT(M2)                                                  PREDD156
      WRITE (6,101) WMACH                                               PREDD157
  101 FORMAT (7H0(PRED),5X,7HWARNING,3X,13HMACH NUMBER =,F8.4,19H IS APPPREDD158
     *ROACHING 1.0)                                                     PREDD159
      MWARN = MWARN + 1                                                 PREDD160
      IF (MWARN .LT. 15) GO TO 23                                       PREDD161
      WRITE (6,102)                                                     PREDD162
  102 FORMAT (7H0(PRED),5X,25H15 WARNINGS HAVE OCCURRED)                PREDD163
      NEXT = .TRUE.                                                     PREDD164
      RETURN                                                            PREDD165
C                                                                       PREDD166
   23 CALL DERV                                                         PREDD167
C                                                                       PREDD168
      RETURN                                                            PREDD169
      END                                                               PREDD170
```

```
CPRINT          MXFLUT OUTPUT ROUTINE
      SUBROUTINE MPRINT
C
      COMMON /CAXIAL/ X,XL,ALOGX
      COMMON /CSPECI/ NSPECI,NF,DX
      COMMON /CREACT/ RHOREA(12)
      COMMON /GASIMW/  TG(2,12,2),ZMWTG(2,12,2),TAU(12,2),CPG(2,12,2)
      COMMON /GASCMP/ YRICH(12),YPRIME(12),FQAIR(2,12,2),ENTH(2,12,2),
     1CONC(16,2,12,2),DUMGAS(216)
      COMMON /JETDAT/ NTUBES,YREACT(12),TS(12),UREACT(12),SPV(12),
     1ZMWT(12),CP(12),FSPECI(12),   G(12),TKE(12),OTHER(36)
      COMMON /CMASS/ ZMASS(12),ZMASSH(12),ZMASDU(24)
      COMMON /CNWNET/ FDUM(100),FUELR(12), FDUMM(100)
C
      DATA JNEW,JRICH,JLEAN /2,1,2/
C
      WRITE (6,1) X
      WRITE (6,2)
      WRITE (6,3)
      WRITE (6,5)
      WRITE (6,4) (YREACT(I),FUELR(I),ZMASSH(I),RHOREA(I),UREACT(I),
     1YRICH(I),FQAIR( JLEAN,I,JNEW),FQAIR(JRICH,I,JNEW),G(I),TAU(I,2),
     2I=1,NSPECI)
C
      RETURN
    1 FORMAT (10X,19HAXIAL DISTANCE, X =,F13.3)
    2 FORMAT (//,10X,6HRADIUS,5X,4HFUEL,8X,4HMASS,5X,7HDENSITY,2X,8HVELO
     1CITY,5X,4HMASS,3X,8HFUEL/GAS,2X,8HFUEL/GAS,7X,1HG,5X,7HTRANSIT)
    3 FORMAT (12X,3H(Y),6X,4HFLOW,8X,4HFLOW,25X,8HFRACTION,3X,4HLEAN,6X
     1,4HRICH,17X,4HTIME)
    4 FORMAT(F20.5,3F10.5,1F10.4,5F10.5)
    5 FORMAT(64X,6H(RICH),33X,5H(SEC))
      END
```

```
CPROPJ        PROPERTIES OF A LAMINAR OR TURBULENT JET
      SUBROUTINE PROPJ(TWO,NTURB,NREG,X,Y,T,TKE,J1,J2)
      COMMON /XPRIN/ OPRIN
      LOGICAL OPRIN
      INTEGER TWO
      LOGICAL NTURB
      REAL MACH,MUL,MUEFF,KCP,MUREF
C*
C*
C*  TWO =1 SINGLE JET
C*  TWO =2 COANNULAR/COPLANAR JET
C*
C*  NTURB=F LAMINAR PROPERTIES ONLY
C*       =T LAMINAR AND TURBULENT PROPERTIES
C*  NREG =1 MIXING REGION (X.LT.XLC)
C*  NREG =2 TRANSITION REGION (X.GE.XLC)
C*  NREG =3 (LARGE X-------- SIMILAR PROFILES)
C*  X   = AXIAL CO-ORDINATE (X/DJET)
C*  Y   = NORMAL CO-ORDINATE,FT.
C*  T   = TEMPERATURE , DEG R
C*  TKE = TURBULENT KE, BTU/LBM
C*
C*  XLC = AXIAL CO-ORDINATE--START OF TRANSITION REGION
C*  MACH = JET DISCHARGE MACH NUMBER
C*
C*
      COMMON/PROPJT/
     * P , PRL , PRT , RGAS , SC ,
     * TREF , MUREF , MACH , XLC ,
     * REFL , C , CHI , RNORM ,
     * RHO(200) , MUL(200) , KCP(200) ,
     * MUEFF(200) , XLN(200) , DK(200) , RETURB(200)
      COMMON/CPROP/CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,CT9,CT10
      COMMON /CPROP2/ CTP, CTS , CTM
      COMMON /PROPJ2/ MACHO,REFLO,YI,YO,MERGE
      LOGICAL MERGE
      REAL MACHO
      COMMON /MISC/ PM(10), DUM33
      LOGICAL LTERP
      COMMON /LLTERP/ LTERP
      COMMON /CTRL/ DUMCL1(9), C6 , DUMCL2(809)
      COMMON /EDGE  / YJETE,SFEDGE
      COMMON /CRMOD / CJRMOD
C*
      DIMENSION Y(1), T(1), TKE(1)
      DATA GCJ/25039.7372/
C*
C*  MUL =LAMINAR VISCOSITY,LBM/FT-SEC
C*  MUEFF =EFFECTIVE VISCOSITY,LBM/FT-SEC
C*  RETURB=TURBULENCE REYNOLDS NUMBER
```

```
C*   KCP =KEFF/CP,LBM/FT-SEC
C*   XLN =TURBULENCE LENGTH SCALE,FT
C*   RHO =DENSITY,LBM/FT3
C*   DK =DIFFUSION PARAMETER FOR TURBULENCE
C*   PR =LAMINAR PRANDTL NUMBER
C*   P   =PRESSURE,PSI
C*   SC =SUTHERLAND CONSTANT,DEG R
C*   TREF = REFERENCE TEMPERATURE,DEG R
C*   MUREF = REFERENCE VISCOSITY,LBM/FT-SEC
C*
C*   RNORM =NORMALIZING TURBULENCE REYNOLDS NUMBER
C*   REFL =REFERENCE DIMENSION FOR TURB. LENGTH SCALE,FT
C*   PRT TURBULENT PRANDTL NUMBER
C*
C*
C*   CONSTANT VALUES
C*
      OPR=1./PRL
      OPRT=1./PRT
      CPRESS=144.*P/RGAS
      IF(NTURB) OREFL=1./REFL
      ORNORM=1./RNORM
      IF(TREF.NE.0.) OTREF=1./TREF
      CKEL=1./1.8
      CM=1./(1.+CT5*MACH)
      GO TO (300,310) , TWO
  310 CMO=1./(1.+CT5*MACHO)
      CT10=2.*CT9-1.
  300 IF(NREG.NE.2) GO TO 33
      XRAT=X/XLC
      CN=CT4*(CT6+CT7*MACH)
      TRANS=XRAT**CN
      IF(.NOT.LTERP) GO TO 3
      CM=1.0/(1.0+CT2*MACH)
      TESTL=CT8*(XRAT-1.)+CT1*CM*(2.-XRAT)
      IF(XRAT  .GE. 2.)  NREG=3
      GO TO 4
    3 CONTINUE
      TESTL=CT3*CM*TRANS
      IF(TESTL .GE. CT8) NREG=3
C
C**   MODIFICATIONS FOR SCALE CHANGE IN CORE REGION
C
   33 YC     = YJETE-REFL/C6
      YINLIM= .9*YC
      YOULIM= YC+.1*REFL/C6
C*
    4 DO 100 J=J1,J2
      TT=T(J)
      RHO(J)=CPRESS/TT
```

```
      VISR=0.
      IF(TREF.NE.0.) GO TO 5
C*
C*  VISCOSITY OF AIR
C*
      MUL(J)=.98292E-6*SQRT(CKEL*TT)/(1.+201.6/TT)
      GO TO 10
C*
C*  LAMINAR VISCOSITY---GENERAL
C*
    5 TR=TT*OTREF
      MUL(J)=MUREF*TR*SQRT(TR)*(TREF+SC)/(TT+SC)
   10 IF(.NOT. NTURB) GO TO 80
C*
C*  CALCULATION OF TURBULENT PARAMETERS
C*
      YT=Y(J)
C*
C*  TURBULENCE SCALE CALCULATION-- CHECK FOR TYPE OF JET
C*
C*
C******* COMPUTATION OF LENGTH SCALE FOR TURBULENCE ******
C*
      GO TO (20,400) , IWO
   20 GO TO (25,35,40),NREG
C*
C*  MIXING LAYER--REFL=WIDTH OF MIXING ZONE
C*
   25 IF( YT.LT.YINLIM ) CSCALE=C6*CJRMOD
      IF( YT.GE.YOULIM ) CSCALE=REFL
      IF((YT.GE.YINLIM) .AND. (YT.LT.YOULIM)) CSCALE=C6-(C6-REFL)*
     * (YT-YINLIM)/(YOULIM-.9*YC)
      XLN(J)= CT1*CM*CSCALE
      GO TO 50
C*
C*  TRANSITION REGION
C*
   35 XLN(J)=TESTL*REFL
      GO TO 50
C*
C*  SPALDING-- SEQUENCE FOR LENGTH SCALE
C*
   40 XLN(J)=CT8*REFL
      GO TO 50
C*
C*  COANNULAR/COPLANAR JET--SET LUS INDICATOR FOR UP/DN-STREAM
C*  OF MERGE POINT
C*
  400 LUS=1
      IF(MERGE) LUS=2
```

```
CPSEQ2        PSEUDO-EQUILIBRIUM COMPOSITION CH(N)-AIR-H2O SYSTEM
C             BRUTE-FORCE METHOD
C
      SUBROUTINE PSEQ2(FAR,WAR,HC,T,BETA,FIXCO,FIXNO,ZZ)
C
      LOGICAL FIXCO,FIXNO,SAMIX,TROUBL,NOSOLN,DONE
      REAL MF,K1,K2,K3,K4,K5
      DIMENSION Z(11),ZZ(11)
      DIMENSION B(3),A(9),DZ(3)
      DOUBLE PRECISION BETA
      COMMON /GMSC  / FF(25),HH(25),SR(25),CPZ(25),DCPR(25)
C
      EQUIVALENCE(DSUMH,B(1)),(DSUMO,B(2)),(DSPM,B(3)),
     1(DZH2,DZ(1)),(DZO2,DZ(2)),(DZH2O,DZ(3))
C
C    ORDER OF SPECIES = H,O,H2,O2,OH,H2O,CO,CO2,N2,A,NO
C
      TROUBL= .FALSE.
      SAMIX=(FAR-OLDFAR+WAR-OLDWAR+HC-OLDHC).EQ.0.0
      IF(SAMIX)GO TO 10
C
C    CALCULATE EQUIVALENCE RATIO
C
      OLDFAR=FAR
      OLDWAR=WAR
      OLDHC=HC
      MF=12.01+1.008*HC
      FARS=0.209495/28.9666*MF/(1.0+0.25*HC)
      ER=FAR/FARS
C
C    TOTAL POUND-ATOMS EACH CONSTITUENT PER POUND MIX
C
      GAR=1.0+FAR+WAR
      SUMH=(FAR*HC/MF+WAR*2.0/18.016)/GAR
      SUMC=(FAR/MF+3E-4/28.9666)/GAR
      SUMO=(WAR/18.016+2.0*(0.209495+3E-4)/28.9666)/GAR
      SUMN=2.0*0.780881/28.9666/GAR
      SUMA=0.009324/28.9666/GAR
      CC=2.0*SUMC+0.5*SUMH-SUMO
      ZZX=0.5*(3E-4+28.9666/18.016*WAR)*SUMN/0.780881
      CCL=SUMC+0.5*SUMH-ZZX
      IF(CCL.LT.0.0)CCL=0.0
      CCR=0.5*(SUMO-SUMC-ZZX)
C
C    UNREACTED, COMPLETELY REACTED, AND SPECIFIED SPECIFIC MOLARITY 1/MWT
C
      SPMU=0.5*(SUMO+SUMC+SUMH+SUMN-ZZX)+SUMA
      SPMC=0.5*(0.5*SUMH+SUMN+SUMO)+SUMA
      IF(ER.GT.1.0)SPMC=0.5*(SUMH+SUMN)+SUMC+SUMA
   10 BETA1=1D0-BETA
```

```
      SPM=BETA1*SPMU+BETA*SPMC
C
C   SET CONCENTRATIONS OF ISOLATED SPECIES, ZERO OTHERS.
C
      CALL SETM(1,0,0,Z,11)
      Z(10)=SUMA
      IF(FIXCO)Z(7)=ZZ(7)/(ZZ(7)+ZZ(8))*SUMC
      IF(FIXNO)Z(11)=ZZ(11)/(ZZ(11)+2.0*ZZ(9))*SUMN
      IF(T.EQ.OLDT)GO TO 20
C
C   GET EQUILIBRIUM CONSTANTS
C
      OLDT=T
      TXK    = T /1.8
      CALL THRM( TXK , 1. )
      K1=EXP(FF(3)+FF(4)-2.0*FF(5))
      K2=EXP(FF(3)+FF(4)-FF(6)-FF(2))
      K3=FF(3)+FF(5)-FF(6)-FF(1)
      IF(K3.GT.88.0)K3=88.0
      K3=EXP(K3)
      K4=EXP(FF(8)+FF(3)-FF(6)-FF(7))
      K5=EXP(FF(9)+FF(4)-2.0*FF(11))
C
C   INITIALIZE PRIMARY SPECIES
C
   20 IF(ER.GT.1.0)GO TO 30
      Z(3)=BETA1*CCL-Z(7)
      IF( Z(3).LT.0. ) Z(3)=0.
      Z(4)=0.5*(Z(3)+Z(7)-Z(11)-CC)
      GO TO 40
   30 Z(4)=BETA1*CCR-0.5*Z(11)
      IF(FIXCO)GO TO 35
      AAA=K4-1.0
      CCC=CC+2.0*Z(4)+Z(11)
      BBB=-(CCC*AAA+K4*SUMC+0.5*SUMH)
      CCC=CCC*K4*SUMC
      Z(7)=-0.5*(BBB+SQRT(BBB**2-4.0*AAA*CCC))/AAA
   35 Z(3)=CC+2.0*Z(4)-Z(7)+Z(11)
   40 Z(6)=SUMO+Z(7)-Z(11)-2.0*(SUMC+Z(4))
C
C   BEGIN ITERATION
C
      DTOL=1E-6
      IF(BETA1.LT.1E-6)DTOL=1E-7
      DO 100 I=1,30
C
C   RADICALS FROM EQUILIBRIA
C
      Z(5)=SQRT(K1*Z(3)*Z(4))
      Z(2)=K2*Z(3)*Z(4)/Z(6)
```

```fortran
      Z(1)=K3*Z(3)*Z(5)/Z(6)
      IF(FIXCO)GO TO 50
      Z(7)=K4*SUMC*Z(3)/(Z(6)+K4*Z(3))
   50 Z(8)=SUMC-Z(7)
      IF(FIXNO)GO TO 60
      ZO2K5=0.25*K5*Z(4)
      Z(11)=SQRT(ZO2K5*(ZO2K5+2.0*SUMN))-ZO2K5
   60 Z(9)=0.5*(SUMN-Z(11))
C
C   H,O CONTINUITY DEFECTS AND MOLARITY DEFECT
C
      DSUMH=SUMH-Z(1)-Z(5)-2.0*(Z(3)+Z(6))
      DSUMO=SUMO-Z(2)-Z(5)-Z(6)-Z(7)-Z(11)-2.0*(Z(4)+Z(8))
      DSPM=SPM
      DO 70 J=1,11
   70 DSPM=DSPM-Z(J)
C
C   TEST FOR CONVERGENCE
C
      DONE=(ABS(DSUMH).LT.DTOL*SUMH)
     1.AND.(ABS(DSUMO).LT.DTOL*SUMO)
     1.AND.(ABS(DSPM).LT.1E-8        )
      IF(DONE) GO TO 110
C
C   PARTIAL DERIVATIVES TO DIRECT CONVERGENCE
C
      Z3=AMAX1(Z(3),1E-36)
      Z4=AMAX1(Z(4),1E-36)
      A(1)=2.0+0.5*(3.0*Z(1)+Z(5))/Z3
      A(2)=(Z(2)+0.5*Z(5))/Z3
      DCODX=K4*SUMC/(Z(6)+K4*Z(3))**2
      IF(.NOT.FIXCO)A(2)=A(2)-DCODX*Z(6)
      A(3)=1.0+(1.5*Z(1)+Z(2)+0.5*Z(5))/Z3
      A(4)=0.5*(Z(1)+Z(5))/Z4
      A(5)=2.0+(Z(2)+0.5*Z(5))/Z4
      DNOD02=0.0
      IF(Z(4).NE.0.0)DNOD02=0.5*K5*Z(9)/(0.25*K5*Z4+Z(11))
      IF(.NOT.FIXNO)A(5)=A(5)+DNOD02
      A(6)=1.0+(Z(2)+0.5*(Z(1)+Z(5)))/Z4
      IF(.NOT.FIXNO)A(6)=A(6)+0.5*DNOD02
      A(7)=2.0-Z(1)/Z(6)
      A(8)=1.0-Z(2)/Z(6)
      IF(.NOT.FIXCO)A(8)=A(8)+DCODX*Z(3)
      A(9)=1.0-(Z(1)+Z(2))/Z(6)
C
C   NEW ESTIMATES OF PRIMARY SPECIES
C
      CALL SOLV3(A,B,DZ,NOSOLN)
      TROUBL=TROUBL.OR.NOSOLN.OR.I.EQ.21
C
```

```
C     PROBLEM DIAGNOSTICS
C
      IF(TROUBL)WRITE(6,1000)I,Z
 1000 FORMAT(1X,I2,8X,  1P11E11.4)
      IF(DONE)GO TO 110
C
      DZLIM = -.99*Z3
      IF(DZH2.LT.DZLIM)DZH2=DZLIM
      Z(3)=Z(3)+DZH2
      DZLIM = -.99*Z4
      IF(DZO2.LT.DZLIM)DZO2=DZLIM
      Z(4)=Z(4)+DZO2
      Z(6)=Z(6)+DZH2O
C
  100 CONTINUE
C
C     PROBLEM DIAGNOSTICS
C
  110 IF(TROUBL)WRITE(6,1001)FAR,WAR,HC,T,BETA,FIXCO,FIXNO
 1001 FORMAT(15H0 I           H,
     '                 10X,1HO,10X,2HH2,9X,2HO2,9X,2HOH,9X,3HH2O,8X,
     '2HCO,9X,3HCO2,8X,2HN2,9X,2HAR,9X,2HNO/
     '25H0PSEQ2 DIAGNOSTICS   FAR=,F9.6,6H  WAR=,F9.6,5H  HC=,E10.3,
     '4H  T=,F9.2,7H  BETA=,D17.10,8H  FIXCO=,L2,8H  FIXNO=,L2)
C
C     COMPOSITION RETURNED AS MOLE FRACTIONS
C
      CALL FMPYC(1,1./SPM,Z,ZZ,11,1)
C
      RETURN
      END
```

```
*QIREM*       QUADRATIC INTERPOLATION ROOT EVALUATION         'QIREM'
C             FOR FUNCTIONS WITH MAXIMUMS
C
      SUBROUTINE QIREM(X,Y, XJP, QV)
       DIMENSION QV(8)
C
C     INPUT-
C      X      = ABSCISSA
C      Y      = ORDINATE (OR ERROR)
C      XJP    = X-JUMP TO BE TAKEN BEFORE ROOT/MAX IS SPANNED,  THE SIGN I
C               A POSITIVE ERROR
C     QV      = STORAGE FOR EIGHT ELEMENT QIRE VECTOR
C     QV(1)  = CTR =0. (FIRST ENTRY ONLY)
C      YTOL   = TOLERANCE ON THE ERROR
C      YO     = ORDINATE TO BE OBTAINED (OPTIONAL)
C      DYDX   = ESTIMATE OF SLOPE FOR 2ND GUESS (OPTIONAL)
C      CTRMAX= MAXIMUM NO. OF ITERATIONS (=25 IF NOT SPECIFIED)
C
C     OUTPUT-
C      X      = NEXT X ESTIMATE
C      QV(1) = 0. IF  YTOL  HAS BEEN SATISFIED
C      QV(5) = 0. IF MAX PT HAS BEEN FOUND WITHIN YTOL,
C                     AND ABS(E).GT.YTOL,
C
C     NOTES-
C      C      = THIRD COEFFICIENT IN THE EQUATION- Y=A+B*X+C*X**2
C             = DI2 IN QIRE NOTATION
C      N1     = EXIT VALUE OF QV(5),  N1=4 IF  X  IS THE PREDICTED MAX PT,
C               N1=+5(-5) IF X IS JUST TO THE LEFT(RIGHT) OF THE PREVIOUSL
C               PREDICTED MAX PT, N1=6 IF X IS THE SECOND PT CLOSE TO THE
C               OTHERWISE N1=N.
C      M      = ENTRY VALUE OF QV(5)
C      SGM    = SIGN OF  M  IF ABS(M)=5
C      SDYDX = SIGN OF THE SLOPE OF THE CURVE
C      XJ     = JUMP TO BE TAKEN FROM LAST  X
C      XJA    = ABSOLUTE VALUE OF MAXIMUM JUMP = ABS(XJP)
C      XM     = DISTANCE FROM CENTRAL PT TO MAX/MIN OF PARABOLA, =XMAX-XX(
C               OR = DISTANCE FROM CENTRAL PT TO THE ROOT,  =XROOT-XX(2)
C      X1     = INPUT (OR LAST) X VALUE
C
      COMMON /CQIREM/ YTOL,YO,DYDX,CTRMAX
      COMMON /ERASE / BOI,C,DXDY,E,I,II,IN,ISPAN,M,N,RADICL,SDYDX,SGN,
     1                TOP,X1,X13,X13P,XJ,XJA,XM, DX(3),DY(3),QV1(10)
       DIMENSION XX(4),YY(4)
       EQUIVALENCE (CTR,QV1(1)), (N1,QIND,QV1(5)),
     1             (XX,QV1(2)), (YY,QV1(6))
C
C     INITIALIZING  AND  PRELIMINARY CHECKING
      IF(CTRMAX.EQ.0.) CTRMAX=25.
      DO 30 I=1,8
```

```
   30  QV1(I)= QV(I)                                                    QIREM-51
       E      = Y-YO                                                    QIREM-52
       M      = N1                                                      QIREM-53
       IF(CTR.EQ.0.) M=0                                                QIREM-54
       SGM    = 1.                                                      QIREM-55
       IF(M.GE.0) GO TO 36                                              QIREM-56
       M      = 5                                                       QIREM-57
       SGM    = -1.                                                     QIREM-58
   36 N       = MIN0(M,3)                                               QIREM-59
C      SDYDX = SIGN(1.,-XJP)                                            QIREM-60
C      (ALTERNATE CALC TO CIRCUMVENT COMPILER ERROR)                    QIREM-61
       IF(XJP) 41,42,42                                                 QIREM-62
   41 SDYDX = 1.                                                        QIREM-63
       GO TO 43                                                         QIREM-64
   42 SDYDX = -1.                                                       QIREM-65
   43 XJA     = ABS(XJP)                                                QIREM-66
       X1     = X                                                       QIREM-67
       IF(M-5) 44,45,46                                                 QIREM-68
   44 IF(ABS(E).LE.YTOL) GO TO 800                                      QIREM-69
       IF(M.EQ.4 .AND. ABS(E-YY(2)).LE.YTOL) GO TO 700                  QIREM-70
       IF(CTR.GE.CTRMAX) CALL ERROR1                                    QIREM-71
       GO TO 50                                                         QIREM-72
   46 M       = 3                                                       QIREM-73
   45 X13P    = XX(3)-XX(1)                                             QIREM-74
C                                                                       QIREM-75
C     DETERMINE INDEX FOR INSERTING CURRENT X,E INTO XX,YY TABLE WHICH ISQIREM-76
C      ORDERED ACCORDING TO X.                                          QIREM-77
   50 IN      = 1                                                       QIREM-78
       IF(N.EQ.0)          GO TO 90                                     QIREM-79
   60  IF(XX(IN).GT.X1) GO TO 70                                        QIREM-80
        IN      = IN+1                                                  QIREM-81
        IF(IN.LE.N)        GO TO 60                                     QIREM-82
       GO TO 90                                                         QIREM-83
C                                                                       QIREM-84
C     RELOCATE IN PREPARATION FOR INSERTING  X,E                        QIREM-85
   70 II      = N+1                                                     QIREM-86
   80  XX(II)= XX(II-1)                                                 QIREM-87
        YY(II)= YY(II-1)                                                QIREM-88
        II      = II-1                                                  QIREM-89
        IF(II.NE.IN) GO TO 80                                           QIREM-90
C                                                                       QIREM-91
C     INSERT NEW POINT                                                  QIREM-92
   90 N       = N+1                                                     QIREM-93
       XX(IN)= X1                                                       QIREM-94
       YY(IN)= E                                                        QIREM-95
C                                                                       QIREM-96
C     LOCATE INTERVAL WHICH SPANS ROOT                                  QIREM-97
       ISPAN = 0                                                        QIREM-98
       IF(N.EQ.1) GO TO 200                                             QIREM-99
        DO 110 I=2,N                                                    QIREM-00
```

```
      IF(SDYDX*YY(I).GT.0. .AND. SDYDX*YY(I-1).LT.0.) ISPAN=I           QIREM-01
  110 CONTINUE                                                          QIREM-02
C                                                                       QIREM-03
C    REDUCE XX,YY TABLE TO THREE POINTS                                 QIREM-04
      IF(N.LE.3)      GO TO 200                                         QIREM-05
      IF(ISPAN.EQ.0) GO TO 140                                          QIREM-06
C     (ROOT HAS BEEN SPANNED)                                           QIREM-07
  122 IF(ISPAN.EQ.N) GO TO 150                                          QIREM-08
      IF(ISPAN.EQ.2) GO TO 175                                          QIREM-09
      IF(ABS(YY(1)).GT.ABS(YY(4))) GO TO 150                            QIREM-10
      GO TO 175                                                         QIREM-11
C                                                                       QIREM-12
C     (ROOT HAS NOT BEEN SPANNED)                                       QIREM-13
  140 IF(IN.LE.2)     GO TO 175                                         QIREM-14
C                                                                       QIREM-15
C     DELETE FIRST POINT                                                QIREM-16
  150 DO 160 I=1,N                                                      QIREM-17
       XX(I) = XX(I+1)                                                  QIREM-18
  160  YY(I) = YY(I+1)                                                  QIREM-19
      ISPAN = ISPAN-1                                                   QIREM-20
C     DELETE FOURTH POINT                                               QIREM-21
  175 N      = N-1                                                      QIREM-22
C                                                                       QIREM-23
C    SIMPLE X-JUMP PREDICTION                                           QIREM-24
  200 N1     = N                                                        QIREM-25
      IF(ISPAN.GT.0 .OR. DYDX.NE.0.) GO TO 205                          QIREM-26
C     XJ     = SDYDX*SIGN(XJA,-E)                                       QIREM-27
C     (ALTERNATE CALC TO CIRCUMVENT COMPILER ERROR)                     QIREM-28
      XJ     = XJP                                                      QIREM-29
      IF(E.LT.0.) XJ=-XJ                                                QIREM-30
      GO TO 900                                                         QIREM-31
C                                                                       QIREM-32
C    CURVE FIT PREDICTIONS                                              QIREM-33
  205 IF(N-2) 210,220,300                                               QIREM-34
C                                                                       QIREM-35
C    ONE POINT PREDICTION BASED ON INPUT VALUE OF DXDY                  QIREM-36
  210 XJ     = -E/DYDX                                                  QIREM-37
      GO TO 900                                                         QIREM-38
C                                                                       QIREM-39
C    TWO POINT STRAIGHT LINE PREDICTION                                 QIREM-40
  220 BOT    = YY(2)-YY(1)                                              QIREM-41
      IF(BOT.EQ.0.) GO TO 230                                           QIREM-42
      DXDY   = (XX(2)-XX(1))/BOT                                        QIREM-43
      IF(DXDY*SDYDX.GT.0.) GO TO 240                                    QIREM-44
C     (CURVE SLOPE IS WRONG - MOVE TOWARD MAXIMUM POINT)                 QIREM-45
  230 XJ     = -3.*SDYDX*XJA                                            QIREM-46
      GO TO 900                                                         QIREM-47
C     (CURVE SLOPE IS CORRECT)                                          QIREM-48
  240 XJ     = -E*DXDY                                                  QIREM-49
      GO TO 900                                                         QIREM-50
```

```
C                                                                       QIREM-51
C     PARABOLIC CURVE FIT PREDICTION                                    QIREM-52
  300 DX(1) = XX(1)-XX(2)                                               QIREM-53
      DX(3) = XX(3)-XX(2)                                               QIREM-54
      DY(1) = YY(1)-YY(2)                                               QIREM-55
      DY(3) = YY(3)-YY(2)                                               QIREM-56
      BOT   = DX(1)*DY(3) - DX(3)*DY(1)                                 QIREM-57
      IF(ABS(BOT).LT.1.E-12) GO TO 600                                  QIREM-58
      TOP   = DX(1)*DX(1)*DY(3) - DX(3)*DX(3)*DY(1)                     QIREM-59
      XM    = .5*TOP/BOT                                                QIREM-60
      X13   = XX(3)-XX(1)                                               QIREM-61
      IF(ABS(XM).GT.ABS(1.E3*X13)) GO TO 600                            QIREM-62
      C     = BOT/(DX(1)*DX(3)*X13)                                     QIREM-63
      RADICL= XM*XM - YY(2)/C                                           QIREM-64
      IF(RADICL.1F.0.) GO TO 360                                        QIREM-65
      SGN   = SIGN(1.,SDYDX*C)                                          QIREM-66
      XM    = XM + SGN*SQRT(RADICL)                                     QIREM-67
      GO TO 890                                                         QIREM-68
C     (IMAGINARY ROOT, HENCE WE ARE LOOKING FOR THE MAXIMUM POINT.      QIREM-69
C     PREDICT MAX PT IF M=3, SELECT PTS ON LEFT/RIGHT SIDE OF PREVIOUSLYQIREM-70
C     PREDICTED PT IF M=4/5)                                            QIREM-71
  360 IF(H-4) 363,364,365                                               QIREM-72
  363 IF(ABS(XM).LT.XJA) N1=4                                           QIREM-73
      GO TO 890                                                         QIREM-74
  364 XJ    = -X13/8.                                                   QIREM-75
      N1    = 5                                                         QIREM-76
      IF(IN.GT.2) GO TO 900                                             QIREM-77
      XJ    = -XJ                                                       QIREM-78
      N1    = -5                                                        QIREM-79
      GO TO 900                                                         QIREM-80
  365 XJ    = SGH*X13P/4.                                               QIREM-81
      N1    = 6                                                         QIREM-82
      GO TO 900                                                         QIREM-83
C                                                                       QIREM-84
C     RETREAT TO LINEAR INTERPOLATION                                   QIREM-85
  600 IF(ISPAN.GT.0) GO TO 122                                          QIREM-86
      GO TO 140                                                         QIREM-87
C                                                                       QIREM-88
C     MAXIMUM FOUND                                                     QIREM-89
  700 QIND  = 0.                                                        QIREM-90
      GO TO 930                                                         QIREM-91
C                                                                       QIREM-92
C     SOLUTION FOUND                                                    QIREM-93
  800 CTR   = 0.                                                        QIREM-94
      GO TO 930                                                         QIREM-95
C                                                                       QIREM-96
C     FINIS                                                             QIREM-97
  890 X1    = XX(2)+XM                                                  QIREM-98
      GO TO 910                                                         QIREM-99
  900 X1    = X1+XJ                                                     QIREM-00
```

```
  910 CALL DTEST                                                        QIREM-01
      X      = AMAX1(XX(1)-XJA,AMIN1(X1,XX(N)+XJA))                     QIREM-02
      CTR    = CTR+1.                                                   QIREM-03
  930 DO 950 I=1,8                                                      QIREM-04
  950 QV(I) = QV1(I)                                                    QIREM-05
  999 RETURN                                                            QIREM-06
      END                                                               QIREM-07
```

```
CRATCON         COMPUTATION OF RATE CONSTANTS                           RATCON01
      SUBROUTINE RATCON(P)                                              RATCON02
      REAL K5,K6,K7,K8,K9,KR5,KR6,KR7,KR8,KR9                           RATCON03
      COMMON /CKINET/ DUM(6),NREAC,                                     RATCON04
     *                RC1(9),RC2(9),RC3(9),TBE(11,5),RCON(7)            RATCON05
      COMMON /PSFQ  / FOA,BETS,TP,X1(1),X2,X3,X4,X5,X6,XK(15)           RATCON06
      COMMON /GHSC  /  F(25),HH(25),SR(25),CPZ(25),DCPR(25)             RATCON07
      COMMON /CKHO2 / K5(1),K6,K7,K8,K9                                 RATCON08
      DIMENSION X(6)                                                    RATCON09
      EQUIVALENCE (X(1),X1(1))                                          RATCON10
C                                                                       RATCON11
C     **REACTIONS**  (1)  H  +OH +M  =  H2O+M                           RATCON12
C                    (2)  H  +H  +M  =  H2 +M                           RATCON13
C                    (3)  H  +O  +M  =  OH +M                           RATCON14
C                    (4)  O  +O  +M  =  O2 +M                           RATCON15
C                    (5)  H  +O2 +M  =  HO2+M                           RATCON16
C                    (6)  H  +HO2    =  OH +OH                          RATCON17
C                    (7)  OH +HO2    =  H2O+O2                          RATCON18
C                    (8)  O  +HO2    =  OH +O2                          RATCON19
C                    (9)  H  +HO2    =  H2 +O2                          RATCON20
C                                                                       RATCON21
C                                                                       RATCON22
C     IF NREAC.LT.5, THE HYDROPEROXYL REACTIONS ARE NOT CONSIDERED      RATCON23
C                                                                       RATCON24
      DATA C2/.01603286/                                                RATCON25
      TK     = TP/1.8                                                   RATCON26
    1 CALL SETM(1,0.,RCON,5)                                            RATCON27
      NRC    = NREAC                                                    RATCON28
      IF( NRC.EQ.5 ) NRC=9                                              RATCON29
C                                                                       RATCON30
      DO 10 L=1,NREAC                                                   RATCON31
      TB     = 0.                                                       RATCON32
      DO 5 K=1,11                                                       RATCON33
    5 TB     = TB+TBE(K,L)*X(K)                                         RATCON34
   10 RCON(L)= TB*RC1(L)*TK**(RC2(L))*EXP(-RC3(L)/(1.98596*TK))         RATCON35
   20 CALL THRM( TK,1. )                                                RATCON36
      RHOM   = 144.*P*C2/(1545.32*TP)                                   RATCON37
      IF( NRC.LT.5 ) RETURN                                             RATCON38
C                                                                       RATCON39
C     SECTION TO CALCULATE HYDROPEROXYL ADJUSTED RCON(5)                RATCON40
C                                                                       RATCON41
      DO 21 L=1,5                                                       RATCON42
      K      = L+4                                                      RATCON43
      TB     = 1.                                                       RATCON44
      IF( L.NE.1 ) GO TO 21                                             RATCON45
      TB     = 0.                                                       RATCON46
      DO 22 J=1,11                                                      RATCON47
   22 TB     = TB+TBE(J,5)*X(J)                                         RATCON48
   21 K5(L)  = TB*RC1(K)*TK**(RC2(K))*EXP(-RC3(K)/(1.98596*TK))         RATCON49
      RCON(5)= K5(1)                                                    RATCON50
```

```
      IF( X1.LE.0. ) GO TO 50                                           RATCON51
      EQ6    = EXP(F(1)+F(13)-2.*F(5))                                  RATCON52
      EQ7    = EXP(F(5)+F(13)-F(6)-F(4))                                RATCON53
      EQ8    = EXP(F(2)+F(13)-F(5)-F(4))                                RATCON54
      EQ9    = EXP(F(1)+F(13)-F(3)-F(4))                                RATCON55
      KR6    = K6/EQ6                                                   RATCON56
      KR7    = K7/EQ7                                                   RATCON57
      KR8    = K8/EQ8                                                   RATCON58
      KR9    = K9/EQ9                                                   RATCON59
      EQ5    = EXP(F(1)+F(4)-F(13))                                     RATCON60
      KR5    = RHOM*K5/(P*EQ5/14.69594)                                 RATCON61
      XNUMER= RHOM*KR5+KR5/K5*(KR6*X5**2/(X1*X4)+KR9*X3/X1+KR7*X6/X1    RATCON62
     * +KR8*X5/X1)                                                      RATCON63
      DENOM = RHOM*(KR5+(K6+K9)*X1+K7*X5+K8*X2)                         RATCON64
      RCON(5)= RCON(5)*(1.-XNUMER/DENOM)                                RATCON65
      IF( RCON(5).LE.0. ) RCON(5)=0.                                    RATCON66
C                                                                       RATCON67
   50 RETURN                                                            RATCON68
      END                                                               RATCON69
```

```
CREADIN        INPUT ROUTINE FOR EMISSIONS PROG.
C            SORT,CONVERT AND STORE
C
      SUBROUTINE READIN
C
      LOGICAL NODATA
      REAL MWT,MATRX,LHV
      DIMENSION AMBT(5),CYCLE(5),FUEL(3),AIDAT(10,11),CARDI(4,11),
     'RADII(11),STDVAL(10),TITLE2(10),ZATR(11)
      COMMON /CBITS / BITS,BLANK
      COMMON /TROUBL/ ERR,ERRMAJ,INERR,PRERR
      LOGICAL         ERR,ERRMAJ,INERR,PRERR
C
      COMMON/INDATA/HC,HF,WAR,TT2,BBETA,TT25,FFAR5,EIC,PP0,
     'RRCO(11),RRCO2(11),RRHC(11),RRNOX(11),PT(11),PS(11),BLOC(11),
     'EICOC(11)
      COMMON/JETDAT/NPTS,RAD(12),TSJ(12),UJ(12),SPV(12),MWT(12),
     'CP(12),FBAR(12),SPLDG(12),OTHER1(12,4)
      COMMON/GASCMP/RICH(12,2),F(2,12,2),H(2,12,2),Z(16,2,12,2),
     'HCINCP(12,2),OTHER2(2,12,2,4)
      COMMON/OPCTRL/TITLE(20),PRINT(30)
C
      EQUIVALENCE(RAD(1),RADII(1)),(P0,AMBT(1)),(T0,AMBT(2)),
     '(HUM,AMBT(3)),(V0,AMBT(4)),(RADJ,AMBT(5)),(T2,CYCLE(1)),
     '(FAR5,CYCLE(2)),(EINO2C,CYCLE(3)),(BETA,CYCLE(4)),(T25,CYCLE(5))
C
      DATA STDVAL/14.696,518.69,0.00551,0.0,1.0,
     '518.69,0.02,20.0,0.0,518.69/
      DATA TITLE2(1)
     '/60HCONSUMPTION OF EMISSIONS IN AUGMENTOR EXHAUST PLUME          /
C
      NAMELIST/ISIDAT/AMBT,P0,T0,HUM,V0,RADJ,
     'CYCLE,T2,FAR5,FINO2C,BETA,T25,FUEL,
     'ALDAT,CARDI,RADII,PT,PS,BLOC,EICOC,
     'TITLE,SF,PRINT
C
C   RESET ALL INPUT VARIABLES TO STANDARD VALUES
C
      CALL SETM(1,BLANK,TITLE,20)
      CALL MOVE(2,STDVAL(1),AMBT,5,1,STDVAL(6),CYCLE,5,1)
      CALL SETM(5,BITS,CARDI,44,RADII,11,PT,44,AIDAT,110,PRINT,30)
      FUEL(1)=2.0
      FUEL(2)=537.0
      FUEL(3)=BITS
      SF=1.0
C
C   READ ENGINE TEST DATA AND CYCLE PARAMETERS
C
      READ(5,ISIDAT)
      IF( INERR ) RETURN
```

```
C                                                                       READIN51
C     MOVE GENERAL DATA INTO COMMON BLOCKS                              READIN52
C                                                                       READIN53
      WAR=HUM                                                           READIN54
      TT2=T2                                                            READIN55
      BBETA=BETA                                                        READIN56
      TT25=T25                                                          READIN57
      FFARS=FARS                                                        READIN58
      FIC=1E-3*EINO2C                                                   READIN59
      PP0=P0                                                            READIN60
      HC=FUEL(1)                                                        READIN61
      LHV=FUEL(3)                                                       READIN62
C                                                                       READIN63
C     ESTIMATE FUEL HEATING VALUE IF NOT GIVEN                          READIN64
C                                                                       READIN65
      IF(LHV.EQ.BITS)LHV=(184686.04+37977.7*HC)/(11.91468+HC)           READIN66
      HF=LHV-1066.4605*(158.1830+47.0276*HC)/(12.01+1.008*HC)           READIN67
      DTF=FUEL(2)-537.0                                                 READIN68
      HF=HF+DTF*(0.43733823+DTF*(3.0451344E-4+DTF*(3.7682468E-8         READIN69
     1+DTF*1.2234568E-10)))                                             READIN70
C                                                                       READIN71
C     SUBSTITUTE CHANGE VECTORS, IF ANY, INTO ALDAT                     READIN72
C     SET STANDARD VALUES IF NO VALUE GIVEN IN INPUT LIST               READIN73
C                                                                       READIN74
      IF(ALDAT(9,1).EQ.BITS) ALDAT(9,1)=0.0                             READIN75
      DO 10 N=1,11                                                      READIN76
      IF(CAROL(1,N).NE.BITS)CALL MOVE(1,CAROL(1,N),ALDAT(1,N),4,1)      READIN77
      IF(ALDAT(1,N).EQ.BITS)GO TO 20                                    READIN78
      IF(RADII(N).NE.BITS)ALDAT(5,N)=RADII(N)                           READIN79
      IF(PI(N).NE.BITS)ALDAT(6,N)=PI(N)                                 READIN80
      IF(PS(N).NE.BITS)ALDAT(7,N)=PS(N)                                 READIN81
      IF(ALDAT(7,N).EQ.BITS)ALDAT(7,N)=P0                               READIN82
      IF(BLOC(N).NE.BITS)ALDAT(8,N)=BLOC(N)                             READIN83
      IF(ALDAT(8,N).EQ.BITS)ALDAT(8,N)=BETA                             READIN84
      IF(EICOC(N).NE.BITS)ALDAT(9,N)=EICOC(N)                           READIN85
      IF(ALDAT(9,N).EQ.BITS)ALDAT(9,N)=ALDAT(9,1)                       READIN86
C                                                                       READIN87
C     MOVE PROBE DATA INTO COMMON BLOCKS                                READIN88
C                                                                       READIN89
      RRCO(N)=SF*ALDAT(1,N)                                             READIN90
      RRCO2(N)=SF*ALDAT(2,N)                                            READIN91
      RRHC(N)=SF*ALDAT(3,N)                                             READIN92
      RRNOX(N)=SF*ALDAT(4,N)                                            READIN93
      PI(N)=ALDAT(6,N)                                                  READIN94
      PS(N)=ALDAT(7,N)                                                  READIN95
      BLOC(N)=ALDAT(8,N)                                                READIN96
      EICOC(N)=ALDAT(9,N)*1E-3                                          READIN97
      RAD(N)=ALDAT(5,N)                                                 READIN98
C                                                                       READIN99
C     SET UP HETEROGENEOUS GAS MODEL                                    READIN00
```

```
C                                                                       READIN01
      CALL MXFLTO(N)                                                    READIN02
   10 NPTS=N                                                            READIN03
C                                                                       READIN04
C   SET UP AMBIENT AIR PROPERTIES                                       READIN05
C                                                                       READIN06
   20 NA=NPTS+1                                                         READIN07
      RAD(NA)=RADJ                                                      READIN08
      TSJ(NA)=T0                                                        READIN09
      UJ(NA)=V0                                                         READIN10
      FBAR(NA)=0.0                                                      READIN11
      SPLDG(NA)=0.0                                                     READIN12
      RICH(NA,1)=0.0                                                    READIN13
      F(1,NA,1)=0.0                                                     READIN14
      F(2,NA,1)=0.0                                                     READIN15
      CALL EQGAST(0.0,WAR,HC,T0,P0,.FALSE.,.FALSE.,ZAIR,                READIN16
     '     HAIR,MWT(NA),SX,SPV(NA),AX,CP(NA))                           READIN17
      CALL SETM(1,0.0,Z(1,1,NA,1),32)                                   READIN18
      CALL SETM(1,HAIR,H(1,NA,1),2)                                     READIN19
      CALL MOVE(2,ZAIR,Z(1,1,NA,1),11,1,ZAIR,Z(1,2,NA,1),11,1)          READIN20
C                                                                       READIN21
C   SET STANDARD PRINT STATIONS IF NOT SPECIFIED                        READIN22
C                                                                       READIN23
      IF(PRINT(1).NE.BITS) GO TO 999                                    READIN24
      PRINT(1)=0.1*RADJ                                                 READIN25
      PRINT(2)=10.0*RADJ                                                READIN26
C                                                                       READIN27
C    GENERATE INPUT FOR JETMIX                                          READIN28
C                                                                       READIN29
  999 CALL JETPRE                                                       READIN30
C                                                                       READIN31
      RETURN                                                            READIN32
      END                                                               READIN33
```

```
CREADT        READS -JETMIX- OUTPUT TAPE
C
C*    OCTOBER 1973  MODIFICATIONS FOR AUG EMIS SYS
C
      SUBROUTINE READT (IENTRY)
C
      COMMON /CBITS / BITS,BLANK
      INTEGER                BLANK
      COMMON /CJETOT/ GCJ,DIAJ,VJET,EJET,NXTA
      COMMON /CSTART/ GJET(200),GEX
      COMMON /CCOUNT/ ZX(5),NSL(5),NX,XMAX
      COMMON /CINPUT/ ZPSI(100,5),ZR(100,5),ZU(100,5),ZE(100,5),
     1  ZRHO(100,5),ZXLN(100,5),ZF(100,5)
      COMMON /CFILK/ CSC
      COMMON /CPRFL / PSI(200),Y(200),UD(200),ED(200),RHO(200),XLN(200),
     * ALX(100,12),NPD,FSPECI(12),NC,X(100)
      REAL MOLF1(100),MOLF2(100),MOLF3(100),MOLF4(100),MOLF5(100),
     1  MOLF6(100),MOLF7(100),MOLF8(100),MOLF9(100),MOLF10(100),
     2  MOLF11(100),MOLF12(100)
      INTEGER TEST,BDY,XXX
      LOGICAL FOUND
      EQUIVALENCE  ( FUEL, FSPECI )
      EQUIVALENCE (MOLF1(1),ALX(1,1)),(MOLF2(1),ALX(1,2)),(MOLF3(1),
     1  ALX(1,3)),(MOLF4(1),ALX(1,4)),(MOLF5(1),ALX(1,5)),(MOLF6(1),
     2  ALX(1,6)),(MOLF7(1),ALX(1,7)),(MOLF8(1),ALX(1,8)),(MOLF9(1),
     3  ALX(1,9)),(MOLF10(1),ALX(1,10)),(MOLF11(1),ALX(1,11)),
     4  (MOLF12(1),ALX(1,12))
C
      COMMON / KEYS / KEYA(11), KEYB(11), KODA(11), KODB(11)
C
      DIMENSION  FUEL(12)
      DIMENSION DUMMY1(2000),DUMMY2(98),DUMMY3(73),DUMMY4(1311),
     * DUMMY5(59)
      DIMENSION THD(200),TID(200),U(200),I(200),TDI(200),
     * XMACH(200),PTOT(200),TTD(200),PTD(200)
      DIMENSION CNAME(12),ALE(12),SCM(12),CPC(36)
C
      DATA     KPREJT/6HPREJET/
      DATA  JETMIX,BDY,TEST,XXX /6HJETMIX,3HBDY,4HTEST,2HXX/
C
      NAMELIST /NLPROF/ NPD,PSI,Y,UD,ED,RHO,XLN,MOLF1,MOLF2,MOLF3,
     1  MOLF4,MOLF5,MOLF6,MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12
      NAMELIST / JETXX / GCJ, DIAJ, VJET, EJET, X, NXTA, NC
      NAMELIST / REFDAT / GJET, FUEL
      NAMELIST /RDATA/ DIAJ,VJET,EJET,GEX,GCJ,NXTA,NC,X
C
C*      SAVE KEYA(4)
C*
      K4SV  = KEYA(4)
C
```

```
      IF (IENTRY.GT.0) GO TO 100
*
      CSC=1000.
C
C*    READ JETMIX DATA FROM TAPE
C*
      KEYA(4) = JETMIX
      KEYA(7)= XXX
      READ (2) KXX1,KREC,
     * DUMMY5,BITS,ERK,GC,GCJ,
     * FOJT,DIAJ,MJET,TJET,PTJET,VJET,TIJET,EJET,PF,VE,ME,TIE,TE,
     * AXI,NJ,NM,UF,MIXPRE,XLC,FLOWJ,MERGE,NV,CON1,
     * CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,CT9,CTP,CTS,CTM,
     * GAM,RG,PR,PRT,SC,TREF,MUREF,SP,SV,SLEN,DPRIN,PLOT,C6,
     * MIX,CF,MAXIT,TOL,SUPB,
     * X,DUMMY1,NXTA,I,NC,DUMMY2
C*
      XMAX=X(NXTA)
      NX=5
      IF (NXTA.LT.5) NX=NXTA
      NXREM=NXTA-NX
C*
C*    READ INITIAL -FAR- AND -G- PROFILES FROM PREJET DATA
C*
      KEYA(4) = KPREJT
      KEYA(7)= BLANK
      READ (1)
     * DIAJ,MJET,TIJET,VJET,PE,TE,TIE,VE,GFX,RG,PR,PRT,SC,TREF,
     * MUREF,DIFF,NC,CNAME,ALE,SCM,CPC,NJ,NM,CT1,CT2,CT3,CT4,CT5,
     * CT6,CT7,CT8,GJET,Y,VD,IHD,IID,ED,
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,
     * DUMMY3,FUEL
C*
C
      NREAD=1
      GO TO 110
  100 IF (NXREM.LE.0) GO TO 230
      IF (NXREM.EQ.1) NX=2
      IF (NXREM.EQ.2) NX=3
      IF (NXREM.EQ.3) NX=4
      NXREM=NXREM-NX+1
      NREAD=2
*
C*    RESET KEYA(4) - - READ REMAINDER OF JETMIX DATA
C*
  110 CONTINUE
      NSPECI = NC
      KEYA(4) = JETMIX
      DO 210 I=NREAD,NX
```

```
      IACT=I+IENTRY                                                     READT101
      ZX(I)=X(IACT)                                                     READT102
      XX=ZX(I)                                                          READT103
      KEYA(7) = CSC * XX + 0.5                                          READT104
      FOUND=.FALSE.                                                     READT105
      READ (2) JREC,KXX,KREG,                                           READT106
     * SUPD,SUPSTP,CORE,CORSTP,MER,MERSTP,NPD,                          READT107
     * PSI,Y,UD,THD,ED,TID,RHD,XLN,U,T,TOT,XMACH,                       READT108
     * PTOT,TTD,PTD,                                                    READT109
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,                             READT110
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,J                         READT111
      IF( KXX.EQ.KEYA(7) ) FOUND=.TRUE.                                 READT112
      IF (.NOT.FOUND) GO TO 210                                         READT113
      NSL(I)=NPD                                                        READT114
      DO 200 J=1,NPD                                                    READT115
      ZPSI(J,I)=PSI(J)                                                  READT116
      ZH(J,I)=Y(J)                                                      READT117
      ZU(J,I)=UD(J)                                                     READT118
      ZE(J,I)=ED(J)                                                     READT119
      ZRHO(J,I)=RHD(J)                                                  READT120
      ZXLN(J,I)=XLN(J)                                                  READT121
      ZF(J,I)=0.                                                        READT122
C*  CALCULATE AVERAGE FUEL/AIR RATIO                                    READT123
      DO 190 K=1,NSPECI                                                 READT124
      ZF(J,I)=ALX(J,K)*FSPECI(K)+ZF(J,I)                                READT125
  190 CONTINUE                                                          READT126
  200 CONTINUE                                                          READT127
  210 CONTINUE                                                          READT128
C*                                                                      READT129
      GEX=0.                                                            READT130
  230 CONTINUE                                                          READT131
C*    RESTORE KEYA(4)                                                   READT132
C*                                                                      READT133
      KEYA(4) = K4SV                                                    READT134
C*                                                                      READT135
      RETURN                                                            READT136
      END                                                               READT137
```

```
*READT2      READS **JETMIX** TAPE (MXFLUT)
C
      SUBROUTINE READT
C
      COMMON /CFILK/ CSC
      COMMON /JMIX1C/ NAME(10),TITLE1(10),IDENT(10),ADDRES(10),
     * IDENT1(10),TMODT(9),ERK,GC,GCJ,FOOT,
     * EJET,AXI,NJ,NM,UE,
     * MIXPRE,XLC,FLOWJ,MERGE,NV,CON1,CT1,CT2,CT3,CT4,CT5,CT6,
     * CT7,CT8,CT9,CTP,CTS,CTM,SC,TREF,MUREF,
     * SX,SV,SLEN,DPRIN,PLOT,C6,MIX,CF,MAXIT,TOL,SUPB,
     * XPRN(100),B(100),JC(100),TC(100),TIC(100),
     * PTC(100),WJ(100),YJ(100),TTC(100),YSONIC(100),YCB(100),
     * XD(100),RD(100),YR(100),YCD(100),PD(100),WV(100),
     * MA2(100),VE2(100),TE2(100),NXTA,NC,CNAME(12),ALJ(12),
     * ALJO(12),ALF(12),SCM(12),CPC(36),DIFF
C
      COMMON /CJMIX2/ SUPD,SUPSTP,CORE,CORSTP,MER,MERSTP,NPD,
     * PSI(200),Y(200),UD(200),ED(200),TID(200),RHO(200),
     * XLN(200),U(200),TOT(200),XMACH(200),PTOT(200),TTD(200),
     * PTD(200),ALX(100,12),T(200)
      COMMON /JETDAT/ NTUBES,YREACT(12),TS(12),UREACT(12),SPV(12),
     1ZMWT(12),CP(12),FSPECI(12),GHAT(12),TKE(12),OTHER(36)
      COMMON /GASCMP/ GASX(1104)
      COMMON /CINPUT/ XY(5),R(5,50),UU(5,50),RRHO(5,50),XSPECI(5,50,13),
     1PPSI(5,50),GG(5,50),NJJ(5),NII
      COMMON /CXLOCA/ XX(5)
      COMMON /CSPECI/ NSPECI,NE,DX
      COMMON /CINPJT/ DIAJ,MJE',TJET,PTJET,VJET,TIJET,PE,VE,ME,TIE,TE,
     1X(100),GAM,RG,PR,PRT
      COMMON /CAXIAL/ XDUM,XL,ALOGX
C
      COMMON /CBITS/ BITS,BLANK
      COMMON / KEYS / KEYA(11), KEYB(11), KODA(11), KODB(11)
      COMMON / FILES / ORGF, UPDF, NEWF, SCRF
      INTEGER     ORGF, UPDF, SCRF
      COMMON / STCTRL / DUMST1(3), FIRSTM, DUMST2(13)
      LOGICAL  FIRSTM
      COMMON /CSPARE/ H(1600),PSIR(50),G(50),SY(50),RAD(50)
      COMMON /CPRINT/ PDUM(20)
      COMMON /XISAVE/ I,ISTAR
C
      DIMENSION SP(13),SPE(12),SPR(12)
C
      REAL MOLF1(100),MOLF2(100),MOLF3(100),MOLF4(100),MOLF5(100),
     1MOLF6(100 )
      REAL         MOLF7(100),MOLF8(100),MOLF9(100),MOLF10(100),
     *             MOLF11(100),MOLF12(100)
      DIMENSION THO(200)
      EQUIVALENCE (MOLF1(1),ALX(1,1)),(MOLF2(1),ALX(1,2)),
```

```
     *              (MOLF3(1),ALX(1,3)),(MOLF4(1),ALX(1,4)),
     *              (MOLF5(1),ALX(1,5)),(MOLF6(1),ALX(1,6)),
     *              (MOLF7(1),ALX(1,7)),(MOLF8(1),ALX(1,8)),
     *              (MOLF9(1),ALX(1,9)),(MOLF10(1),ALX(1,10)),
     *              (MOLF11(1),ALX(1,11)),(MOLF12(1),ALX(1,12))
      LOGICAL FOUND
      INTEGER TEST,BDY,XXX,BLANK
C
      DATA  JETMIX,BDY,TEST,XXX /6HJETMIX,3HBDY,4HTEST,2HXX/
      DATA ISPALD /6HSPALDG/
      DATA SPE/12*-1./
C
      NAMELIST /NLPROF/ UD,MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,
     1 RHO,XLN,Y,U,T,XMACH,NPD,P3I
      NAMELIST /JETXX/ VJET,X,DIAJ,NC
      NAMELIST /NLSPLD/ G,SY,RAD,JEND
C
      NREACT=NTUBES
      K4SV  = KEYA(4)
      IF ( .NOT. FIRSTM ) GO TO 90
C
      CALL REDPRE
      CALL NEWFLO
      FIRSTM= .FALSE.
C
      NJ=0
C
      CALL SETM(1,BITS,X,50)
      NPD=1
      KEYA(4) = JETMIX
      KEYA(7)= XXX
      READ (2) KXX1,KREC,
     * NAME,TITLE1,IDENT,ADDRES,IDENT1,TWODT,BITS,ERK,GC,GCJ,
     * FOOT,DIAJ,MJET,TJET,PTJET,VJET,TIJET,EJET,PE,VE,MF,TIE,TE,
     * AXI,NJ,NM,UF,MIXPRE,XLC,FLOWJ,MERGE,NV,CON1,
     * CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8,CT9,CTP,CTS,CTM,
     * GAM,RG,PR,PRT,SC,TREF,MUREF,SX,SV,SLEN,DPRIN,PLOT,C6,
     * MIX,CF,MAXIT,TOL,SUPR,
     * X,XPRN,B,UC,TC,TIC,PTC,WJ,YJ,TIC,YSONIC,
     * YCB,XD,RD,YR,YCD,PD,WV,MA2,VE2,TE2,NXTA,I,
     * NC,CNAME,ALJ,ALJD,ALE,SCM,CPC,DIFF,CSC
C*
      DO 80 KK=2,50
      IF( X(KK).EQ.BITS ) GO TO 85
      IF( X(KK).EQ.X(KK-1) ) GO TO 84
   80 CONTINUE
   84 IK    = KK+1
      DO 86 JK=IK,50
   86 X(JK-1)=X(JK)
```

```
   85 CONTINUE
C*
      NSPECI= NC
      NTUBES= NSPECI-1
      NF    = NSPECI+1
      I=1
      ISTAR=1
      NII=5
   90 IF (ISTAR.GT.NII) I=NII
  100 XX(I)=X(ISTAR)
      IF (XX(I).EQ.BITS) GO TO 210
C*  HAVE REACHED END OF STORAGE ON JETMIX TAPE
C
      XY(I)=ALOG(XX(I)+1.)
C*  --ISTAR-- KEEPS TRACK OF CURRENT POSITION IN JETMIX X-VECTOR
C*  --I     -- IS THE LOCATION OF DATA IN STORAGE IN MXFLUT
      KEYA(4)= JETMIX
      CSC=1000.
      KEYA(7) = CSC * XX(I) + 0.5
      FOUND=.FALSE.
      READ (2) JREC,KXX,KREG,
     * SUPD,SUPSTP,CORF,CORSTP,MER,MERSTP,NPD,
     * PSI,Y,UD,THD,ED,TID,RHD,XIN,U,T,TOT,XMACH,
     * PTOT,TTD,PTD,
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,J
       IF( KXX.EQ.KEYA(7) ) FOUND=.TRUE.
      IF (.NOT.FOUND) GO TO 210
      IF( PDUM(15).NE.0. ) WRITE (6,700) X(ISTAR)
  700 FORMAT (//10X,2H**,3X,13HRESTART AT X=,F12.6)
      KEYA(4)= ISPALD
      READ (3) KXX,SY,RAD,G,JEND
C
      NJ    = MIN0(50,NPD )
      PSIR(1)=0.
C
      UCONST= .02
      UEDGE = UCONST*(U(1)-U(NPD) ) + U(NPD)
      JINTER=NPD
  136 IF( U(JINTER).GE.UEDGE ) GO TO 140
      JINTER=JINTER-1
      IF (JINTER.EQ.2) GO TO 140
      GO TO 136
  140 NINTER=NPD-JINTER+1
      CALL LSPFIT(U(JINTER),PSI(JINTER),NINTER,UEDGE,PSIEDG,1,0)
      DPSI=(PSIEDG-PSI(1))/FLOAT(NJ-1)
      IF( PDUM(15).NE.0. ) WRITE (6,138) UEDGE,PSIEDG,NINTER,JINTER,
     *                                   XX(I)
  138 FORMAT (1X,27HUEDGE,PSIEDG,NINTER,JINTER,1HX,2F10.6,2I3,F10.6)
C*  RESTORE DATA ON ZERO-TO-ONE- BASIS (PERCENT JET WIDTH)
```

```
      CALL NEWPSI( PSIR,NJ )
      CALL FMPYC( 1,PSIEDG,PSIR,PSIR,50 )
      DO 151 J=1,NPD
  151 Y(J)  = Y(J)**2
      CALL LSPFIT (PSI,Y,NPD,PSIR,H,NJ,0)
      CALL LSPFIT (PSI,U,NPD,PSIR,H(NJ+1),NJ,0)
      CALL LSPFIT (PSI,RHO,NPD,PSIR,H(2*NJ+1),NJ,0)
C
      DO 160 K=1,NSPECI
      KP2NJ1=(K+2)*NJ+1
      CALL LSPFIT (PSI,ALX(1,K),NPD,PSIR,H(KP2NJ1),NJ,0)
      NPDM1=NPD-1
C
C*  CHECK ON JETMIX CALCULATIONS
      SP(K)=0.
      DO 155 J=1,NPDM1
  155 SP(K)=SP(K)+(ALX(J+1,K)+ALX(J,K))*(PSI(J+1)-PSI(J))*.5
  160 CONTINUE
C
C*  STORE **G** RESULTS FROM--SPLDG--
      KP2NJ1= (NSPECI+3)*NJ+1
      CALL LFIT  (SY,G,JEND,PSIR,H(KP2NJ1),NJ,0)
C
      SP(NF)=0.
      DO 161 K=1,NSPECI
      IF (SPE(K).LT.0.) SPE(K)=SP(K)
      SPR(K)=SP(K)/SPE(K)
  161 SP(NF)= SP(NF)+SP(K)
      IF( PDUM(15).NE.0. ) WRITE (6,2) SP,SPR
    2 FORMAT (1X,29HSPEICE CONSERVATION IN JETMIX,/,8H  SPECIE,8F12.8,/,
     18H   RATIO,8F12.4)
      PSISCL= VJET*DIAJ**2/144.*3.1415927
      DIAJO2= .5*DIAJ
      NJJ(I)=NJ
      DO 200 J=1,NJ
      PPSI(I,J)= PSIR(J)*PSISCL
      R(I,J)= DIAJO2**2*H(J)
      NJP1=NJ+J
      UU(I,J)=H(NJP1)
      NJX2P1=NJ*2+J
      RRHO(I,J)=H(NJX2P1)
      XSPECI(I,J,NF)=0.
C
C*  CALCULATE AVERAGE MIXTURE RATIO
      DO 190 K=1,NSPECI
      KP2NJJ=(K+2)*NJ+J
      XSPECI(I,J,K)=H(KP2NJJ)
  190 XSPECI(I,J,NF)=XSPECI(I,J,K)*FSPECI(K)+XSPECI(I,J,NF)
      KP2NJ1= (NSPECI+3)*NJ+J
      GG(I,J)=H(KP2NJ1)
```

```
  200 CONTINUE                                                         READT201
      ISTAR=ISTAR+1                                                    READT202
      I=ISTAR                                                          READT203
      IF (I.LE.NII) GO TO 100                                          READT204
      GO TO 220                                                        READT205
C                                                                      READT206
C*  NEARING END OF JETMIX RUN                                          READT207
  210 NII=NII-1                                                        READT208
      IF (XL.LE.X(ISTAR-1)) GO TO 220                                  READT209
      IF( PDUM(15).NE.0. ) WRITE (6,1) XL,X(ISTAR-1)                   READT210
      XL=X(ISTAR-1)                                                    READT211
C                                                                      READT212
  220 KEYA(4) = K4SV                                                   READT213
      RETURN                                                           READT214
C                                                                      READT215
    1 FORMAT (10X,20(2HA*),//      ,10X,71HREQUESTED END VALUE OF X IS BEYREADT216
     1OND END OF JETMIX RUN, XL HAS BEEN RESET,/,20X,17HOLD VALUE OF XL READT217
     1=,F10.6,/,20X,17HNEW VALUE OF XL =,F10.6)                        READT218
      END                                                              READT219
```

```
CREDPRE     -REDPRE- SUB. TO READ PREJET DATA
C
C     NOV, 1973
C
      SUBROUTINE REDPRE
C
      COMMON /INDATA/ N,HF,WAR,T2,BETA,T25,FARS,EINO2C,P0,
     *                RCO(11),RCO2(11),RHC(11),RNOX(11),
     *                PT(11),PS(11),BLOC(11),QCO(11)
      REAL            N
      COMMON /JETDAT/ NPTS,RAD(12),TS(12),UJ(12),SPV(12),MWT(12),
     *                CP(12),FULL(12),SPALDG(12),TKE(12),OTHER1(36)
      REAL            MWT
      COMMON / GASCMP / RICH(24), FUUL(48), ENTH(48),
     *                  CONC(768), HCINCP(24), OTHER2(192)
      COMMON /OPCTRL/ TITLE(20),PRINT(30)
      COMMON /GASTMW/ TG(2,12,2),DUMTMW(120)
C
      COMMON / KEYS / KEYA(11), KEYB(11), KODA(11), KODB(11)
      DIMENSION GJET(200),Y(200),UD(200),THD(200),TID(200),ED(200),
     * MOLF1(100),MOLF2(100),MOLF3(100),MOLF4(100),
     * MOLF5(100),MOLF6(100),MOLF7(100),MOLF8(100),MOLF9(100),
     * MOLF10(100),MOLF11(100),MOLF12(100)
      DIMENSION CNAME(12),ALE(12),SCM(12),CPC(36)
C
      DIMENSION H(2,12,2)
      EQUIVALENCE (H(1,1,1),ENTH(1))
      DATA       KPREJT/6HPREJET/
C
      NAMELIST / REFDAT / NPTS, RAD, TS, U, SPV, MWT, CP,
     1 FUEL,SPALDG,TKE,OTHER1,
     2 RICH,FUUL,ENTH,CONC,HCINCP,OTHER2
     * ,TITLE,PRINT
     * ,N,HF,WAR,T2,BETA,T25,FARS,EINO2C,P0,RCO,RCO2,RHC,RNOX,
     * PT,PS,BLOC,QCO
C
      K4SV  = KEYA(4)
      KEYA(4) = KPREJT
C
      READ (1)
     * DIAJ,MJET,TIJET,VJET,PE,TF,TIF,VE,GFX,RG,PR,PRT,SC,TREF,
     * MUREF,DIFF,NC,CNAME,ALE,SCM,CPC,NJ,NM,CT1,CT2,CT3,CT4,CT5,
     * CT6,CT7,CT8,GJET,Y,UD,THD,TID,ED,
     * MOLF1,MOLF2,MOLF3,MOLF4,MOLF5,MOLF6,
     * MOLF7,MOLF8,MOLF9,MOLF10,MOLF11,MOLF12,
     * NPTS,RAD,TS,UJ,SPV,MWT,CP,FUEL,SPALDG,TKE,OTHER1,
     * TITLE,PRINT,N,HF,WAR,T2,BETA,T25,FARS,EINO2C,P0,
     * RCO,RCO2,RHC,RNOX,PT,PS,BLOC,QCO,
     * RICH,HCINCP,FUUL,ENTH,CONC,OTHER2
     * ,GEX
```

```
C                                                                       REDPRE51
      KEYA(4) = K4SV                                                    REDPRE52
C                                                                       REDPRE53
C     INITIALIZE RICH FRACTION TEMPERATURE                              REDPRE54
C                                                                       REDPRE55
      DO 10 KK=1,NPTS                                                   REDPRE56
      CALL TFMH1( CONC,H(1,KK,1),TG(1,KK,1),CP)                         REDPRE57
   10 CONTINUE                                                          REDPRE58
C                                                                       REDPRE59
      RETURN                                                            REDPRE60
C                                                                       REDPRE61
      END                                                               REDPRE62
```

```
CRH2         HYDROGEN CONCENTRATION ESTIMATED FROM CO MEASUREMENT        RH200001
C            REF LTR GLEASON TO STABRYLLA 3.8/71                         RH200002
C                                                                        RH200003
      FUNCTION RH2(RCO1)                                                 RH200004
C                                                                        RH200005
      IF(RCO1.EQ.RCO)GO TO 30                                            RH200006
C                                                                        RH200007
      RCO=RCO1                                                           RH200008
      IF(RCO.GT.0.091022292)GO TO 20                                     RH200009
C                                                                        RH200010
      H2CO=0.2*(100.0*RCO)**0.16                                         RH200011
      GO TO 30                                                           RH200012
C                                                                        RH200013
   20 H2CO=0.0147*(100.0*RCO)**1.342                                     RH200014
C                                                                        RH200015
   30 RH2=H2CO*RCO                                                       RH200016
      RETURN                                                             RH200017
      END                                                                RH200018
```

```
CSCALE          TURBULENCE SCALE/SINGLE,COANNULAR/COPLANAR JETS              SCALE001
      SUBROUTINE SCALE(UU,THOJ,NREG,X)                                       SCALE002
C****** CONTROL COMMON                                                       SCALE003
C*                                                                           SCALE004
      COMMON /CTRL/                                                          SCALE005
     * NXTA      , CMPRS      , QJET       , TURBJ      , COEF(10)   ,       SCALE006
     * NPU       , NPD        , DXC        , XU         , XDD        ,       SCALE007
     * DSTOR(800)                                                            SCALE008
C*                                                                           SCALE009
C****** PROFILE COMMON                                                       SCALF010
C*                                                                           SCALE011
      COMMON /PROF/ PST(200),Y(200),UD(200),THD(200),ED(200)                 SCALE012
C****** CONSTANT AND ERROR COMMON                                            SCALE013
C*                                                                           SCALE014
      COMMON /CNERR/ BITS , ERR , GC , GCJ , FOOT                            SCALE015
C*                                                                           SCALE016
C****** BOUNDARY CONDITION COMMON                                            SCALE017
C*                                                                           SCALE018
      COMMON /BC/ UEDGE  , EEDGE  , THEDGE                                   SCALE019
C*                                                                           SCALE020
      COMMON /BC0/  U0, E0, TH0                                              SCALE021
      COMMON /CTRL2/                                                         SCALE022
     * EDGEI  , SFI    , MERGE  , XMERGE , YMERGE ,                          SCALE023
     * SLOPEI , SLOPED , CEPTI  , CEPTO                                      SCALE024
      LOGICAL MERGE                                                          SCALE025
      DIMENSION UU(1)                                                        SCALE026
      INTEGER THOJ                                                           SCALE027
      COMMON /EDGE/  YJETE,SFEDGE                                            SCALE028
      COMMON /PROPJ2/  MACHO,REFLO,YI,YO,MERGP                               SCALE029
      LOGICAL MERGP                                                          SCALE030
      COMMON /PROPJT/                                                        SCALE031
     * P        , PRL      , PRT      , RGAS     , SC       ,                SCALE032
     * TREF     , MUREF    , MACH     , XLC      ,                           SCALE033
     * REFL     , C        , CHI      , RNORM    ,                           SCALE034
     * RHO(200)     , MUL(200)     , KCP(200)     ,                          SCALE035
     * MUEFF(200) , XLN(200)     , DK(200)      , RETURB(200)                SCALE036
      COMMON /CPROP/ CT1,CT2,CT3,CT4,CT5,CT6,CT7,CT8                         SCALE037
C*                                                                           SCALE038
      COMMON /MIXER/ MIX,RD(100),XD(100),CF,YR(100)                          SCALE039
      LOGICAL  MIX                                                           SCALE040
      COMMON /FLOBAL/ MAXIT,SUPR,NIT,PSID,YDD,YDC,                           SCALE041
     *               P1,P2,UCL,TOL,UPSTRM,CVG                                SCALE042
      LOGICAL  SUPR,CVG,UPSTRM                                               SCALE043
      COMMON /ACONVG/ YCD(100),PD(100),INDE(100),  CHOKE, CHOKED             SCALF044
      LOGICAL  CHOKE, CHOKED                                                 SCALE045
      COMMON /DEIT/  CLSP(100)                                               SCALE046
      COMMON /STA2/ MACH2,TS2,SS2,V2,RHO2,DPDX2                              SCALE047
      REAL  MACH2                                                            SCALF048
      COMMON /BCMIX2/ GRADU,TW,MUW,RHOW,PTF,TTE                              SCALE049
      REAL  MUW                                                              SCALE050
```

```
      COMMON /CBODY/  YCB(100),CLSPCB(100),YCB1 , UCL1
      COMMON /OUTMIX/ NXORIG
      COMMON /SCALED/ SCLD,ALXLIM
      LOGICAL         SCLD
C*
      EQUIVALENCE  (C6,COEF(6))
C*
    1 GO TO  (100,200)  ,  TWOJ
C*
C*  SINGLE JET-- COMPUTE LOCAL WIDTH OF MIXING ZONE
C*
  100 GO TO  (140,120,120) , NREG
  120 REFL=C6*YJETE
      GO TO 160
  140 IF( SCLD ) GO TO 142
      DO 143 L=1,NPU
      IF(UU(L).NE.UCL1)  GO TO 144
  143 CONTINUE
      GO TO 144
  142 DO 1442 L=1,NPU
      IF( UU(L).GE.ALXLIM ) GO TO 144
 1442 CONTINUE
  144 REFL=C6*(YJETE-Y(L-1))
C*
  160 RETURN
C*
C*
C*  COANNULAR/COPLANAR JET
C*
C*  TEST MERGE TO DETERMINE IF UP/DN-STREAM OF MERGE STATION
C*
  200 TEST1=U0-1.E-6
      TEST2=U0+1.E-6
      IF(MERGE)  GO TO 260
C*
C*  UPSTREAM-- COMPUTE  REFL,REFLO,YI,YO
C*
  210 YO=C6*YJETE
      YI=C6*EDGEI
C*
C*  SCAN UU TABLE FOR BOUNDARIES OF MIXING ZONES
C*
      DO 220 L=1,NPU
      IF(UU(L).NE. UCL1)  GO TO 222
  220 CONTINUE
  222 LK=L-1
      REFL=YI-C6*Y(LK)
      DO 225 L=LK,NPU
      IF(UU(L).GT.TEST1  .AND.  UU(L).LT. TEST2)  UU(L)=U0
      IF(UU(L).EQ.U0)  LJ=L
```

```
  225 CONTINUE
C*
  230 REFLO=YO-C6*Y(LJ)
      GO TO 500
C*
C*  DOWN STREAM-- DETERMINE BOUNDARIES OF MIXING ZONES//
C*  USE LINEAR EQUATIONS FOR NOZZLE/MERGE POINT LINES
  260 YI=(SLOPEO*X+CEPTO)  *C6
      YO=C6*YJETF
      REFLO=YO-C6*(SLOPEI*X+CEPTI)
      DO 270 L=1,NPU
      IF(UU(L).NE. 1.)  GO TO 275
  270 CONTINUE
C*
  275 REFL=YI-C6*Y(L-1)
C*
  500 RETURN
      END
```

```
CSCKP          MAIN DRIVER FOR SCKP CALCULATION
      SUBROUTINE SCKP( KK )
      COMMON /GASCMP/ RICH(12,2),FUEL(2,12,2),ENTH(2,12,2),
     *                CONC(16,2,12,2),HCINCP(12,2),OTHER(192)
      COMMON /GASTMW/ TG(2,12,2),MWTG(2,12,2),TAU(12,2),CPG(2,12,2)
      REAL            MWTG
      COMMON /PSFQ  / FOA,BFTS,TP,X(16),DHQDMW,TEQ,BFQ,XMWT,HNEQ
      COMMON /CBITS / BITS,BLANK
      COMMON /CKINET/ DUMC1(4),XMOLWO,DUMSK(90),ENTRY1
      LOGICAL         ENTRY1
      COMMON /INMOLF/ XIN(12)
      COMMON /INDATA/ HOC1,HF,WAR,T2,BTA,T25,FARS,FINO2C,PO,DUM2(88)
      COMMON /PSFQX / HOC,HUM,CO2AIR,MAIR,FS,FUELMW
      REAL                        MAIR
      COMMON /SNMW  / ALSP(150), WMT(75)
      COMMON /GHSC  / FF(25),HZ(25),DUMG(75)
      COMMON /CPRINT/ PDUM(20)
      DIMENSION C1(16),C2(16)
      DATA RO/1.98596/
C
      HOC    = HOC1
      HUM    = WAR
      FUELMW= 12.01+1.008*HOC
      FS     = FUELMW/(1.+.25*HOC)*.209495/MAIR
      K      = KK
      WRITE (6,100) K
  100 FORMAT(//8X,10HSTREAMTUBE,2X, I2//)
      TP     = TG(1,K,2)
      IF( TP.LE.1500. ) GO TO 4
      TPK    = TP/1.8
      CALL THRM( TPK,1. )
      CALL SETM(2,0.,C1,16,C2,16)
      Y2     = 0.
      DO 90 I=12,16
   90 Y2     = Y2+WMT(I)*CONC(I,1,K,2)
      QY2    = 0.
      IF( Y2.EQ.0. ) GO TO 92
      QY2    = 1./Y2
   92 QYY2   = 1./(1.-Y2)
C
C     SPLIT MIXTURE TO CONSERVE ENERGY
C
      CALL FMPYC(1,QY2,CONC(12,1,K,2),C2(12),5)
      CALL FMPYC(1,QYY2,CONC(1,1,K,2),C1,11)
      HS1    = 0.
      HS2    = 0.
      TMOLES= 0.
      DO 91 I=1,16
      TMOLES= TMOLES+C1(I)
      HS1    = HS1+C1(I)*HZ(I)
```

```
      HS2    = HS2+C2(I)*HZ(I)
   91 CONTINUE
      HS1    = RO*TP*HS1
      HS2    = RO*TP*HS2
      QTMOL  = 1./TMOLES
      CALL FMPYC(1,QTMOL,C1,XIN,11)
      H      = HS1
      IF( PDUM(10).NE.0. ) CALL TABPRT( 5HXINIT,XIN,11,10)
C
C   CALCULATE EQUIVALENCE RATIO
C
      ZETA1  = 2.*XIN(9)+XIN(11)
      ZETA   = XIN(8)-.5*CO2AIR/.780881*ZETA1
      ER     = 3.*.780881/.209495*(XIN(7)+ZETA)/ZETA1
      FOA1   = FS*ER
      DTIME  = TAU(K,2)-TAU(K,1)
C
C   CALL FOR KINETICS STEP
C
      ENTRY1= .TRUE.
      XMOLWO= BITS
    2 CALL KINET(PO,H,FOA1,0.)
C
      CALL KINET(PO,H,FOA1,DTIME)
C
C   STORE THE NEW CONCENTRATIONS
C
      IF( PDUM(10).NE.0. ) CALL TABPRT(6HT.XFIN,TP,12,10)
    3 TERM   = 1./(XMWT*(1.-Y2))
      CALL FMPYC(1,TERM,X,CONC(1,1,K,1),11)
      CALL FMPYC(1,Y2,C2(12),CONC(12,1,K,1),5)
      ENTH(1,K,1)= HS1*(1.-Y2)+HS2*Y2
      TG(1,K,1)= TP
      GO TO 5
    4 TG(1,K,1)= TG(1,K,2)
      ENTH(1,K,1)= ENTH(1,K,2)
      CALL MOVE(1,CONC(1,1,K,2),CONC(1,1,K,1),16,1)
C
    5 RETURN
      END
```

```
CSERCH       OPTIMAL SEQUENTIAL SEARCH TECHNIQUE
      SUBROUTINE SEARCH (F,FOFX,A,X,B)
C
C    THIS ROUTINE USES AN OPTIMAL SEQUENTIAL SEARCH TECHNIQUE TO FIND X
C     IN (A,B) SUCH THAT F(X) = FOFX
C
C    ASSUMPTIONS
C     (1) F(X) CONTINUOUS ON THE CLOSED INTERVAL (A,B)
C     (2) FOFX IS NOT EQUAL TO ZERO
C
      COMMON/SINT/DUM(10),KOUNT,FX
C
      X1 = (2.*A + B)/3.
      KOUNT = 1
      F1 = F(X1)
      IF (ABS(1.-F1/FOFX) .GT. 0.0001) GO TO 2
      X = X1
      FX = F1
      RETURN
C
    2 X2 = (A + 2.*B)/3.
      KOUNT = KOUNT + 1
      F2 = F(X2)
      IF (ABS(1.-F2/FOFX) .GT. 0.0001) GO TO 3
      X = X2
      FX = F2
      RETURN
C
    3 D1 = F1 - FOFX
      IF (D1*(FOFX-F2) .GT. 0.) GO TO 5
      IF (D1 .LT. 0.) GO TO 4
      X2 = A
      KOUNT = KOUNT + 1
      F2 = F(X2)
      IF (F2 .LT. FOFX  .AND.  ABS(1.-F2/FOFX) .GT. 0.0001) GO TO 5
      X = A
      FX = F2
      RETURN
C
    4 X1 = B
      KOUNT = KOUNT + 1
      F1 = F(X1)
      IF (F1 .GT. FOFX  .AND.  ABS(1.-F1/FOFX) .GT. 0.0001) GO TO 5
      X = B
      FX = F1
      RETURN
C
    5 X = (X1 + X2)/2.
      KOUNT = KOUNT + 1
      FX = F(X)
```

```
      IF (ABS(1.-FX/FOFX) .LE. 0.0001) RETURN                          SERCH051
      IF (KOUNT .EQ. 11) RETURN                                         SERCH052
      IF ((FX-FOFX)*(FOFX-F2) .LT. 0.) GO TO 6                          SERCH053
      X1 = X                                                            SERCH054
      GO TO 5                                                           SERCH055
    6 X2 = X                                                            SERCH056
      F2 = FX                                                           SERCH057
      GO TO 5                                                           SERCH058
C                                                                       SERCH059
      END                                                               SERCH060
```

```
CSETUP        SETS-UP FLOW PROPERTIES FOR INTERPOLATION
C
      SUBROUTINE SETUP (IRST,K)
C
      COMMON /CPROLI/ G(50,1),ALPHA(1,1),BETA(50,1),GAMM(50,1),
     1  DELTA(50,1)
      COMMON /CINIT/ XX,DPSI(1)
      COMMON /CSTART/ GJET(200),GEX
      COMMON /CCOUNT/ X(5),NSL(5),NX,XMAX
      COMMON /CINPUT/ PSI(100,5),R(100,5),U(100,5),E(100,5),RHO(100,5),
     1  XLN(100,5),F(100,5)
      COMMON /CNUNET/ PSIMAX(5),NSLZ(5),RZ(50,5),UZ(50,5),EZ(50,5),
     1  RHOZ(50,5),XLNZ(50,5),FZ(50,5)
      COMMON /CTRANS/ RZT(5,50),UZT(5,50),EZT(5,50),RHOZT(5,50),
     1  XLNZT(5,50),FZT(5,50)
      COMMON /CSLDAT/ SY(50),RAD(50),VEL(50),TKE(50),DEN(50),TLS(50),
     1  FAR(50)
      COMMON /CADCNT/ MAX,NS1,NXS,NSZMAX,NSZ1
C*
C*  DETERMINE VALUES OF STREAM FUNCTION ON STREAMLINES AT INPUT STATIONS
      DO 170 I=1,NX
      NS=NSL(I)
      PSIMAX(I)=PSI(NS,I)
      SLZ=PSIMAX(I)/DPSI(1)
      NSLZ(I)=IFIX(SLZ)+1
      IF (NSLZ(I).GT.MAX) GO TO 180
  170 CONTINUE
      NXS=NX
      GO TO 190
  180 NXS=I
      NSLZ(NXS)=MAX
  190 SY(1)=0.
      NSZMAX=NSLZ(NXS)
      DO 200 J=2,NSZMAX
      SY(J)=SY(J-1)+DPSI(1)
  200 CONTINUE
C*  CALCULATE FLUID PROPERTIES ON STREAMLINES AT INPUT STATIONS
      DO 205 I=1,NXS
      NS=NSL(I)
      NSZ=NSLZ(I)
      CALL LFIT1 (PSI(1,I),R(1,I),NS,SY,RZ(1,I),NSZ,0)
      CALL LFIT1 (PSI(1,I),U(1,I),NS,SY,UZ(1,I),NSZ,0)
      CALL LFIT1 (PSI(1,I),E(1,I),NS,SY,EZ(1,I),NSZ,0)
      CALL LFIT1 (PSI(1,I),RHO(1,I),NS,SY,RHOZ(1,I),NSZ,0)
      CALL LFIT1 (PSI(1,I),XLN(1,I),NS,SY,XLNZ(1,I),NSZ,0)
      CALL LFIT1 (PSI(1,I),F(1,I),NS,SY,FZ(1,I),NSZ,0)
  205 CONTINUE
      DO 215 IR=1,NXS
      I=NXS+1-IR
      NSZ=NSLZ(I)
```

```
      NSZP1=NSLZ(I)+1
      IF (NSZP1.GT.NSZMAX) GO TO 215
      DO 210 J=NSZP1,NSZMAX
      RZ(J,I)=RZ(J,I+1)*RZ(NSZ,I)/RZ(NSZ,I+1)
      UZ(J,I)=U(NS,I)
      EZ(J,I)=E(NS,I)
      RHOZ(J,I)=RHO(NS,I)
      XLNZ(J,I)=XLN(NS,I)
      FZ(J,I)=F(NS,I)
  210 CONTINUE
  215 CONTINUE
      IF (IRST.GT.0) GO TO 220
      NSZ1=NSLZ(1)
      IF (UZ(NSZ1,1).LT.(0.0)) UZ(NSZ1,1)=0.
      IF (EZ(NSZ1,1).LT.(0.0)) EZ(NSZ1,1)=0.
      IF (FZ(NSZ1,1).LT.(0.0)) FZ(NSZ1,1)=0.
      CALL LFIT1 (PSI,GJET,NS1,SY,G,NSZ1,0)
      IF (G(NSZ1,K).LT.(0.0)) G(NSZ1,K)=0.
  220 CONTINUE
C*   TRANSPOSE MATRICES FOR INTERPOLATION ALONG STREAMLINES
      DO 240 I=1,NXS
      DO 230 J=1,NSZMAX
      RZT(I,J)=RZ(J,I)
      UZT(I,J)=UZ(J,I)
      EZT(I,J)=EZ(J,I)
      RHOZT(I,J)=RHOZ(J,I)
      XLNZT(I,J)=XLNZ(J,I)
      FZT(I,J)=FZ(J,I)
  230 CONTINUE
  240 CONTINUE
      RETURN
      END
```

```
CSOLV3       SOLVE 3 LINEAR EQUATIONS  A(I,J)*X(J)=B(I)
      SUBROUTINE SOLV3(A,B,X,NOSOLN)
      LOGICAL NOSOLN
      DIMENSION A(3,3),B(3),X(3),D(3,3)
      C=0.0
      DO 20 I=1,3
      DO 10 J=1,3
      I1=I+1
      IF(I1.GT.3)I1=I1-3
      I2=I+2
      IF(I2.GT.3)I2=I2-3
      J1=J+1
      IF(J1.GT.3)J1=J1-3
      J2=J+2
      IF(J2.GT.3)J2=J2-3
   10 D(I,J)=A(I1,J1)*A(I2,J2)-A(I1,J2)*A(I2,J1)
   20 C=C+D(I,3)*A(I,3)
      NOSOLN=C.EQ.0.0
      IF(NOSOLN)RETURN
      DO 40 J=1,3
      X(J)=0.0
      DO 30 I=1,3
   30 X(J)=X(J)+D(I,J)*B(I)
   40 X(J)=X(J)/C
      RETURN
      END
```

```
CSPALD      -SPALD-  MAIN SUBROUTINE FOR CALC. OF -G-
C
C*    OCTOBER 1973  MODIFICATIONS FOR AUG EMIS SYS
C*
      SUBROUTINE SPALD
C
C*  DIAJ = JET DIAMETER -- IN
C*  VJET = JET VELOCITY -- FT/SEC
C*  EJET = JET TKE -- BTU/LBM
C*  GJET = INITIAL DISTRIBUTION OF -G- IN JET
C*  GEX = VALUE OF -G- IN EXTERNAL FLOW
C*  GCJ = DIMENSIONAL CONSTANT -- LBM-FT**2/BTU-SEC**2
C*  PSI = STREAM FUNCTION -- LBM/FT**3
C*  R = DIMENSIONLESS RADIUS
C*  U = DIMENSIONLESS VELOCITY
C*  E = DIMENSIONLESS TKE
C*  F = MEAN FUEL/AIR RATIO
C*  RHO = DENSITY -- LBM/FT**3
C*  XLN = TURBULENCE LENGTH SCALE -- FT
C
      LOGICAL PRINT
      COMMON /CTHETA/ THETA,II
      COMMON /CENDS/ JSTART,JEND
      COMMON /CPBOLI/ G(50,1),ALPHA(1,1),BETA(50,1),GAMM(50,1),
     1  DELTA(50,1)
      COMMON /CTRIDI/ COEFL(50),COEFC(50),COEFR(50),RHS(50)
      COMMON /CBNDRY/ DX,DXPRN,JT
      COMMON /CINIT/ XX,DPSI(1)
      COMMON /CJETOT/ GCJ,DIAJ,VJET,EJET,NXTA
      COMMON /CSTART/ GJET(200),GEX
      COMMON /CCOUNT/ X(5),NSL(5),NX,XMAX
      COMMON /CINPUT/ PSI(100,5),R(100,5),U(100,5),E(100,5),RHO(100,5),
     1  XLN(100,5),F(100,5)
      COMMON /CCONST/ CONST1,CONST2,CONST3,CONST4
      COMMON /CNUNET/ PSIMAX(5),NSLZ(5),RZ(50,5),UZ(50,5),EZ(50,5),
     1  RHOZ(50,5),XLNZ(50,5),FZ(50,5)
      COMMON /CTRANS/ RZT(5,50),UZT(5,50),FZT(5,50),RHOZT(5,50),
     1  XLNZT(5,50),EZT(5,50)
      COMMON /CSLDAT/ SY(50),RAD(50),VEL(50),TKE(50),DEN(50),TLS(50),
     1  FAR(50)
      COMMON /CADCNT/ MAX,NS1,NXS,NSZMAX,NSZ1
      COMMON /CPRFL / DUMM(2414),XW(100)
      COMMON / CFILK / CSC
C
      COMMON / KEYS / KEYA(11), KEYB(11), KODA(11), KODB(11)
C
      DIMENSION  LGSPG(5), NSSPG(6), NS2SPG(2)
C
      DATA      LGSPG/2, 50,  1, 50 ,0/
      DATA      NSSPG/1, 4HIPSI,  1, 2HIR,  1, 2HIG/
```

```
      DATA      NS2SPG/1, 3HIGK/
C
      NAMELIST /CHANGE/ ALFA,CG1,CG2,SIGMAG,THETA,DX,DXPRN,XSTOP,NSZ1
     *                 ,PRINT
C
C
      PRINT = .FALSE.
      XSTOP = 1000.
      DX    = 1000.
      READ (5,CHANGE)
C
      IENTRY=0
      CALL READT (IENTRY)
      IENTRY=IENTRY+NX-1
      JSTART=1
      II=1
      JT=1
      K=1
      IX=1
      IRST=0
      MAX=50
      ALPHA(1,1)=1./(VJET*DIAJ/12.)
      CONST1=ALFA*SQRT(GCJ*EJET)
      CONST2=CG1/(VJET*DIAJ/12.)
      CONST3=CG2*(DIAJ/12.)/(ALFA*ALFA*VJET)
      CONST4=1./SIGMAG
      IF (XSTOP.GT.XMAX) XSTOP=XMAX
C*  SET INITIAL STREAMLINE SPACING
      NS1=NSL(1)
      DO 150 J=1,NS1
      IF (R(J,1).LT.1.0) GO TO 150
      DPSI(1)=PSI(J,1)/(FLOAT(NSZ1-1))
      GO TO 160
  150 CONTINUE
      DPSI(1)=PSI(NS1,1)/(FLOAT(NSZ1-1))
C*  SET-UP FLOW PROPERTIES FOR INTERPOLATION ALONG STREAMLINES
  160 CALL SETUP (IRST,K)
C*  PRINT -G- PROFILE AT INITIAL STATION
      XX=0.
      JEND=NSZ1
      CALL STPRUP (K)
      KEYB(7)= CSC*XX+0.5
      WRITE (3) KXX,SY,RAD,G,JEND
      IF( .NOT.PRINT ) GO TO 170
      CALL TABPRT (6HA*  X=,XX,IX,1)
      CALL TABPRT (5H  PSI,SY,JEND,10)
      CALL TABPRT (3H  R,RAD,JEND,10)
      CALL TABPRT (3H  G,G,JEND,10)
C*  MOVE DOWNSTREAM FROM INITIAL STATION
  170 SYMAX = 0.
```

```
      SYNEXT=SY(NSZ1)+DPSI(1)
C*   DO-LOOP FOR DOWNSTREAM CALCULATIONS
      XD      = XW(2)
      DX      = AMIN1(DX,XD-XW(1))
      IP      = 2
  180 XX      = XX+DX
      ISPACE=0
C*
C*
C*   CHECK TO SEE IF ADDITIONAL INTERPOLATION STATIONS ARE REQUIRED
      IF (XX.LE.X(NX)) GO TO 270
C*   IF REQUIRED, ADD DOWNSTREAM INTERPOLATION STATIONS
      CALL MOVE(5,X(5),X(1),1,1,NSL(5),NSL(1),1,1,PSI(1,5),PSI(1,1),
     * 100,1,R(1,5),R(1,1),100,1,U(1,5),U(1,1),100,1)
      CALL MOVE(4,E(1,5),E(1,1),100,1,RHO(1,5),RHO(1,1),100,1,
     * XLN(1,5),XLN(1,1),100,1,F(1,5),F(1,1),100,1)
      CALL SETM(5,0.,X(2),4,NSL(2),4,PSI(1,2),400,R(1,2),400,U(1,2),400)
      CALL SETM(5,0.,E(1,2),400,RHO(1,2),400,XLN(1,2),400,F(1,2),400,
     * PSIMAX,5)
      CALL SETM(5,0.,RZ,250,UZ,250,EZ,250,RHOZ,250,XLNZ,250)
      CALL SETM(5,0.,FZ,250,RZT,250,UZT,250,EZT,250,RHOZT,250)
      CALL SETM(2,0.,XLNZT,250,FZT,250)
      CALL SETM(1,0,NSLZ,5)
      CALL READT (IENTRY)
      IENTRY=IENTRY+NX-1
C*   SET-UP FLOW PROPERTIES FOR INTERPOLATION THROUGH NEW STATIONS
      IRST=IRST+1
      CALL SETUP (IRST,K)
  270 CONTINUE
C*
C*
C*   CHECK TO SEE IF ANOTHER STREAMLINE SHOULD BE ADDED
      CALL LFIT1 (X,PSIMAX,NX,XX,SYMAX,IX,0)
      IF (SYMAX.LT.SYNEXT) GO TO 320
C*   IF REQUIRED, ADD ANOTHER STREAMLINE
      JEND=JEND+1
C*   CHECK TO SEE IF NUMBER OF STREAMLINES EXCEEDS LIMIT
      IF (JEND.LE.MAX) GO TO 300
C*   IF REQUIRED, RE-SET STREAMLINE SPACING
      ISPACE=1
      DPSI(1)=DPSI(1)+DPSI(1)
      CALL SETM(5,0.,RZ,250,UZ,250,EZ,250,RHOZ,250,XLNZ,250)
      CALL SETM(5,0.,FZ,250,RZT,250,UZT,250,EZT,250,RHOZT,250)
      CALL SETM(5,0.,XLNZT,250,FZT,250,SY,50,RAD,50,VEL,50)
      CALL SETM(5,0.,TKE,50,DEN,50,TLS,50,FAR,50,BETA,50)
      CALL SETM(2,0.,GAMM,50,DELTA,50)
C*   SET-UP FLOW PROPERTIES FOR INTERPOLATION ALONG REVISED STREAMLINES
      IRST=IRST+1
      CALL SETUP (IRST,K)
      NEWJN=(JEND+1)/2
```

```
      DO 290 J=1,NEWJN
      JJ=(2*J)-1
      IF (JJ.GE.JEND) GO TO 280
      G(J,K)=G(JJ,K)
      GO TO 290
  280 G(J,K)=GEX
  290 CONTINUE
      NSTART=NEWJN+1
      NG=MAX-NEWJN
      CALL SETM (1,0.0,G(NSTART,K),NG)
      JEND=NEWJN
      GO TO 310
  300 G(JEND,K)=GEX
      JSTART=JEND
C*  RE-CALC. PDE COEFFICIENTS AT PREVIOUS STATION ON REVISED STREAMLINES
  310 XX=XX-DX
      CALL STPROP (K)
      CALL COEFF(K)
      JSTART=1
      XX=XX+DX
      SYNEXT=SY(JEND)+DPSI(1)
  320 CONTINUE
C*
C*
C*  CALCULATE FLUID PROPERTIES ON ALL STREAMLINES AT NEW STATION
      CALL STPROP (K)
C*  SOLVE PARABOLIC PDE AT EACH GRID POINT AND PRINT RESULTS
      CALL PROLIC(K)
      IF( XX.LT.XD ) GO TO 340
C*    WRITE DATA RECORDS - - AT EACH XX - -
C*                      SY - - RAD - - G
C*
      LGSPG(2) = JEND
      LGSPG(4) = JEND
      KEYB(7) = CSC * XX + 0.5
      KXX    = KEYB(7)
      WRITE (3) KXX,SY,RAD,G,JEND
C
      IF( .NOT.PRINT ) GO TO 330
      WRITE (6,321) XX
  321 FORMAT(/2X,2HX=,F16.6/)
      WRITE (6,322)
  322 FORMAT(//5X,3HPSI,9X,1HR,11X,1HG,11X,3HPSI,9X,1HR,11X,1HG/)
      DO 323 I=1,JEND,2
      WRITE (6,324) SY(I),RAD(I),G(I,1),SY(I+1),RAD(I+1),G(I+1,1)
  323 CONTINUE
  324 FORMAT(2(F10.6,F12.6,E13.4))
  330 IP     = IP+1
      IF( IP.GT.NXTA ) GO TO 400
      XD     = XW(IP)
```

```
      DX      = DXSAVE                                                  SPALD201
C*  RE-SET AXIAL SPACING WHEN STREAMLINE SPACING HAS BEEN RE-SET        SPALD202
  340 IF( ISPACE.EQ.1 ) DX=DX+DX                                        SPALD203
      DXSAVE = DX                                                       SPALD204
      DX      = AMIN1( DX, XD-XX )                                      SPALD205
      GO TO 180                                                         SPALD206
  400 RETURN                                                            SPALD207
      END                                                               SPALD208
```

```
CSTPROP      CALCULATES FLUID PROPERTIES AT EACH POINT                  STPROP01
*                                                                       STPROP02
      SUBROUTINE STPROP (K)                                             STPROP03
*                                                                       STPROP04
      COMMON /CENDS/ JSTART,JEND                                        STPROP05
      COMMON /CINIT/ XX,DPSI(1)                                         STPROP06
      COMMON /CCOUNT/ X(5),NSL(5),NX,XMAX                               STPROP07
      COMMON /CTRANS/ RZT(5,50),UZT(5,50),EZT(5,50),RHOZT(5,50),        STPROP08
     1  XLNZT(5,50),FZT(5,50)                                           STPROP09
      COMMON /CSTDAT/ SY(50),RAD(50),VEL(50),TKE(50),DEN(50),TLS(50),   STPROP10
     1  FAR(50)                                                         STPROP11
      COMMON /CADCNT/ MAX,NS1,NXS,NSZMAX,NSZ1                           STPROP12
*                                                                       STPROP13
      IX=1                                                              STPROP14
      DO 330 J=JSTART,JEND                                              STPROP15
      CALL LFIT1 (X,RZT(1,J),NXS,XX,RAD(J),IX,0)                        STPROP16
      CALL LFIT1 (X,UZT(1,J),NXS,XX,VEL(J),IX,0)                        STPROP17
      CALL LFIT1 (X,EZT(1,J),NXS,XX,TKE(J),IX,0)                        STPROP18
      CALL LFIT1 (X,RHOZT(1,J),NXS,XX,DEN(J),IX,0)                      STPROP19
      CALL LFIT1 (X,XLNZT(1,J),NXS,XX,TLS(J),IX,0)                      STPROP20
      CALL LFIT1 (X,FZT(1,J),NXS,XX,FAR(J),IX,0)                        STPROP21
      IF (VEL(J).LT.(0.0)) VEL(J)=0.0                                   STPROP22
      IF (TKE(J).LT.(0.0)) TKE(J)=0.0                                   STPROP23
      IF (FAR(J).LT.(0.0)) FAR(J)=0.0                                   STPROP24
  330 CONTINUE                                                          STPROP25
      RETURN                                                            STPROP26
      END                                                               STPROP27
```

```
CSUMCPD          DIFFUSION FLUX TERM FOR ENERGY EQUATION
      FUNCTION  SUMCPD(L)
C***********************************************************************
C********************* SPECIAL VERSION FOR 12 SPECIES *****************
C***********************************************************************
      COMMON /ERASE/ DUMS1(400), T(200), DUMS2(200)
      COMMON /DIFEQI/
     * NC , CNAMF(12) , ALJ(12) , ALJO(12) ,ALE(12) , SCM(12) ,
     * TCPRF(12) , HCPRF(12) , CPC(3,12)
      COMMON /MOLUP/ ALXU(1200),DALXU(100,12),DUME(200)
      DIMENSION  CPT(12)
C*
C*
C*
    1 TEMP=T(L)
      DO 10 K=1,NC
   10 CPT(K)=CPC(1,K)+CPC(2,K)*TEMP+CPC(3,K)*TEMP*TEMP
C*
C*
C*   MULTIPLY BY GRADIENTS/SCHMIDT NUMBERS AND SUM
C*
   12 SUMCPD=0.
      DO 15 K=1,NC
   15 SUMCPD=SUMCPD+CPT(K)/SCM(K)*DALXU(L,K)
   20 RETURN
      END
```

```
CSUMP        CALC. AND PRINT RESIDUAL EMISSIONS INDICES
C            AT SELECTED AXIAL POSNS.
C
      SUBROUTINE SUMUP
C
      REAL MWT(2),MWTBAR,M(2)
      INTEGER DAY,YEAR
      DIMENSION SPV(2),CP(2),X(16),T(2),OPT(10,11,3),SUM(4),GAM(2),
     1FGAM(2)
C
      COMMON/GASCMP/YY(12,2),F(2,12,2),H(2,12,2),Z(16,2,12,2),
     * DUM1(24),DUM2(192)
      COMMON/OPCTRL/TITLE(20),PRINT(30)/JETDAT/NT,RAD(12),DUM3(132)
      COMMON/CAXIAL/DIST,DUM9(2)
      COMMON /CMASS / DUM5(12),W(12),DUM6(24)
      COMMON/INDATA/HC,HF,WAR,DUM7(5),P0,DUM8(88)
      COMMON /STCTRL / LSTA,DUMST(16)
      COMMON /CBITS/ BITS,BLANK
C
C
      AVG(X1,X2)=YYY*X1+(1.0-YYY)*X2
C
      CALL SETM(1,BITS,OPT,330)
      CALL SETM(1,0.0,SUM,4)
      DO 70 N=1,NT
C
C   COMPUTE PROPERTIES OF RICH (1) AND LEAN (2) PARTS
C
      DO 20 IP=1,2
      MWT(IP)=0.0
      DO 10 I=1,16
   10 MWT(IP)=MWT(IP)+Z(I,IP,N,1)
      MWT(IP)=1.0/MWT(IP)
      CALL MOVE(1,Z(1,IP,N,1),X,16,1)
      CALL TEMHI(X,H(IP,N,1),T(IP),CP(IP))
      SPV(IP)=10.732459/MWT(IP)*T(IP)/P0
      GAM(IP)=CP(IP)*MWT(IP)/1.98596
   20 GAM(IP)=GAM(IP)/(GAM(IP)-1.0)
C
C   FREE STREAM MEAN PROPERTIES
C
      ASSIGN 30 TO MEAN
      KK=2
      YYY=YY(N,1)
C
   30 FBAR=AVG(F(1,N,1),F(2,N,1))
      SPVBAR=AVG(SPV(1),SPV(2))
      MWTBAR=1.0/AVG(1.0/MWT(1),1.0/MWT(2))
      DO 40 I=1,16
   40 X(I)=AVG(Z(I,1,N,1),Z(I,2,N,1))
```

```
      FICO=28.01*X(7)/FBAR
      FIHC=(120.1+10.08*HC)*X(16)/FBAR
      FINOX=46.008*(X(11)+X(14))/FBAR
      CALL FMPYC(1,MWTBAR,X,X,16)
      DRYMOL=1.006202*(1.0-X(6))
      IF(DRYMOL.GT.1.0)DRYMOL=1.0
      RCO=X(7)/DRYMOL
      RHC=10.0*X(16)
      RNOX=X(11)+X(14)
C
C   PLACE PROPERTIES IN OUTPUT VECTOR
C
      OPT(1,N,KK)=YYY
      OPT(2,N,KK)=FBAR*(1.0+WAR)/(1.0-FBAR)
      OPT(3,N,KK)=1E3*EICO
      OPT(4,N,KK)=1E3*EIHC
      OPT(5,N,KK)=1E3*EINOX
      OPT(6,N,KK)=1E6*RCO
      OPT(7,N,KK)=1E6*RHC
      OPT(8,N,KK)=1E6*RNOX
      GO TO MEAN,(50,70)
C
C   STREAMTUBE AREA AND VELOCITY
C
   50 AREA  = 3.1415927*(RAD(N+1)**2-RAD(N)**2)/144.
      VEL=W(N)*SPVBAR/AREA
      OPT(1,N,1)= RAD(N)/12.
      OPT(2,N,1)= RAD(N+1)/12.
      OPT(3,N,1)=W(N)
      OPT(4,N,1)=VEL
      OPT(5,N,1)=F(1,N,1)
      OPT(6,N,1)=T(1)
      OPT(7,N,1)=F(2,N,1)
      OPT(8,N,1)=T(2)
C
C   FUEL AND CONTAMINANT SUMMATIONS
C
      FFLOW=FBAR*W(N)
      SUM(1)=SUM(1)+FFLOW
      SUM(2)=SUM(2)+FFLOW*EICO
      SUM(3)=SUM(3)+FFLOW*EIHC
      SUM(4)=SUM(4)+FFLOW*EINOX
C
C   MEAN IMPACT PRESSURE
C
      TAU=YY(N,1)*SPV(1)/SPV(2)/(1.0+YY(N,1)*(SPV(1)/SPV(2)-1.0))
      DO 60 IP=1,2
      FGAM(IP)=SQRT(GAM(IP)*(2.0/(GAM(IP)+1.0))**((GAM(IP)+1.0)
     */(GAM(IP)-1.0)))
   60 M(IP)=VEL/SQRT(49723.928*GAM(IP)/MWT(IP)*T(IP))
```

```
      CTAU=1.0-TAU                                                      SUMP0101
      PT=P0*(TAU**2*PRAT(M(1),GAM(1))+CTAU**2*PRAT(M(2),GAM(2)))        SUMP0102
     1/(TAU**2+CTAU**2)                                                 SUMP0103
C                                                                       SUMP0104
C     GAS SAMPLE RICH FRACTION                                          SUMP0105
C                                                                       SUMP0106
      RW1W2=SQRT((T(2)+VEL**2/50075.458/CP(2))                          SUMP0107
     1/(T(1)+VEL**2/50075.458/CP(1))*MWT(1)/MWT(2))                     SUMP0108
      RW1W2=RW1W2*FGAM(1)/FGAM(2)*PRAT(M(1),GAM(1))/PRAT(M(2),GAM(2))   SUMP0109
      YYS=TAU*RW1W2/(TAU*RW1W2+CTAU)                                    SUMP0110
C                                                                       SUMP0111
C     APPARENT MEAN TEMPERATURE (ASPIRATED THERMOCOUPLE)                SUMP0112
C                                                                       SUMP0113
      TTC=(TAU*(T(1)/T(2))**0.45*RW1W2**0.6*T(1)+CTAU*T(2))             SUMP0114
     1/(TAU*(T(1)/T(2))**0.45*RW1W2**0.6+CTAU)                          SUMP0115
      OPT(9,N,1)=PT                                                     SUMP0116
      OPT(10,N,1)=TTC                                                   SUMP0117
C                                                                       SUMP0118
C     APPARENT MEAN PROPERTIES OF GAS SAMPLE                            SUMP0119
C                                                                       SUMP0120
      ASSIGN 70 TO MEAN                                                 SUMP0121
      KK=3                                                              SUMP0122
      YYY=YYS                                                           SUMP0123
      GO TO 30                                                          SUMP0124
C                                                                       SUMP0125
   70 CONTINUE                                                          SUMP0126
C                                                                       SUMP0127
C     OVERALL EMISSIONS INDICES                                         SUMP0128
C                                                                       SUMP0129
      EICO=1E3*SUM(2)/SUM(1)                                            SUMP0130
      EIHC=1E3*SUM(3)/SUM(1)                                            SUMP0131
      EINOX=1E3*SUM(4)/SUM(1)                                           SUMP0132
C                                                                       SUMP0133
C     PRINT RESULTS                                                     SUMP0134
C                                                                       SUMP0135
      CALL CLOCK(TIME,ELAPSD,MONTH,DAY)                                 SUMP0136
      DATA YEAR/75/                                                     SUMP0137
      WRITE (6,1000) TIME,MONTH,DAY,YEAR,ELAPSD,TITLE,PRINT(LSTA),      SUMP0138
     1((OPT(I,J,1),I=1,10),J=1,11)                                      SUMP0139
      WRITE(6,1001)                                                     SUMP0140
      WRITE(6,1002)((OPT(I,J,2),I=1,10),J=1,11)                         SUMP0141
      WRITE(6,1003)                                                     SUMP0142
      WRITE(6,1002)((OPT(I,J,3),I=1,10),J=1,11)                         SUMP0143
      WRITE(6,1004)W(NT+1),SUM(1),EICO,EIHC,EINOX,(SUM(I),I=2,4)        SUMP0144
C                                                                       SUMP0145
C     PUNCH CARDS FOR SUBSEQUENT PLOTTING                               SUMP0146
C     DATA FOR EACH TUBE - INNER, OUTER RADIUS, SAMPLE FUEL/AIR,        SUMP0147
C        EICO, EIHC, EINOX, PPM CO, HC, NOX                             SUMP0148
C                                                                       SUMP0149
      PUNCH 1005,TITLE,PRINT(LSTA),NT,((OPT(I,J,1),I=1,2),(OPT(I,J,3),  SUMP0150
```

```
     *I=2,8),J=1,NT)
C
      RETURN
C
 1000 FORMAT(1H1,12X,40HANALYTICAL MODEL OF EXHAUST PLUMES FROM ,
     *56HAFTERBURNING ENGINES TO PREDICT REACTION OF CONTAMINANT ,
     *10HEMISSIONS./52H GENERAL ELECTRIC COMPANY            ,
     *21HAIRCRAFT ENGINE GROUP,9X,11HCOMPUTED AT,F6.2,10H HOURS ON ,
     *2(I2,1H/),I2,7X,12HELAPSED TIME,F8.2,8H SECONDS/ 6X,20A6//
     *34X,34HPROFILES AND CONTAMINANT RESIDUALS,F8.3,
     *22H FEET FROM NOZZLE EXIT//9X,12HRADIUS, FEET,9X,13HGAS FLOW      ,
     *8HVELOCITY,11X,9HRICH PART,17X,9HLEAN PART,12X,10HPREDICTED ,
     *10HINDICATION/11H        INNER,8X,5HOUTER,9X,3HPPS,10X,3HFPS,7X,
     *2(26HFUEL/GAS      TEMP,DEGR     ),21H PT,PSIA        TT,DEGR//
     *11(1X,1P10E13.5/))
 1001 FORMAT(/28X,42HAVERAGE GAS COMPOSITION UNDER FREE STREAM ,
     *10HCONDITIONS)
 1002 FORMAT(51H0  MASS FRACT     FUEL/AIR          EMISSIONS INDICES, ,
     *11HLB/KLB FUEL,13X,23HANALYZER READINGS, PPMV/13H    RICH PART,
     *6X,5HRATIO,9X,2HCO,11X,2HHC,7X,27HNOX AS NO2    CO(DRIED)     ,
     *9HHC AS CHN,7X,3HNOX//11(1X,1P10E13.5/))
 1003 FORMAT(/28X,39HAVERAGE GAS COMPOSITION AS MODIFIED BY ,
     *12HSAMPLE PROBE)
 1004 FORMAT(/28X,41HINTEGRATED PROFILE AVERAGES (FREE STREAM ,
     *11HCONDITIONS)/23H0        TOTAL FLOWS, PPS,9X,17HEMISSIONS INDICES,
     *13H, LB/KLB FUEL,13X,22HCONTAMINANT FLOWS, PPS/12H      GAS MIX,
     *7X,7HFUEL   ,2(7X,2HCO,11X,2HHC,7X,10HNOX AS NO2)//1X,1P10E13.5)
 1005 FORMAT(2(6X,10A6/),17X,2HX=,F7.3,3H FT,17X,1H(,I2,7H TUBES)/
     *(1P6E12.4/36X,3E12.4))
C
      END
```

```
CTABPRT         TABLE PRINTOUT
      SUBROUTINE TABPRT(NAME,A,NA,NCOL1)
       DIMENSION A(10)
C
C    INPUT-
C     NAME  = ARRAY NAME TO BE PRINTED
C     A     = ARRAY TO BE PRINTED
C     NA    = NUMBER OF ELEMENTS
C     NCOL1 = NUMBER OF COLUMNS TO BE USED IN THE PRINT FORMAT (MAXIMUM
C     I1TAB = LOCATION OF FIRST ELEMENT IN A-ARRAY TO BE PRINTED
C
      COMMON /CBITS / BITS,BLANK
      COMMON /CTABPR/ I1TAB
C
      EQUIVALENCE     (IB,B), (IC,C), (LSPACE,ASPACE)
      DIMENSION FMT(12)
      REAL  I12
C
      DATA IBCI/0010000000000/
      DATA ZERO/0./
      DATA FMT(1)/67H(1X,I5
     1               )/
      DATA
     A F1,        F3,        F6,        E5,        BCD,       OCT,      I12/
     B6H,F12.1, 6H,F12.3, 6H,F12.6, 6H,E12.4, 6H,6X,A6, 6H,8X,O4, 4H,I12
     C/
C
      NCOL  = MIN0(NCOL1,10)
      NB    = NA
C
C    WRITE HEADING
      WRITE (6,1000) NAME
C
   45 I1    = I1TAB
      I     = I1
      I2    = 0
C
C    WRITE LINE SPACE
   47 WRITE (6,1002)
C    LOCATION OF NEXT LINE SPACE IS GIVEN BY THE VALUE OF A(I+1)
      ASPACE= A(I+1)
      IF(LSPACE.LE.1 .OR. LSPACE.GE.IBCI) LSPACE=IBCI
      LSPACE= LSPACE+I-1
      GO TO 110
C
C    BEGIN LOOP TO DEFINE LINE FORMAT
   48 II    = 1
C
C    CHECK FOR NORMALIZED FLOATING NUMBER
   50 B     = A(I)
```

```
      C     = B+ZERO                                                    TABPRT51
      IF(B.EQ.BITS) GO TO 85                                            TABPRT52
      IF(IC.NE.IR) GO TO 80                                             TABPRT53
      IF(B.EQ.0.) GO TO 65                                              TABPRT54
      B     = ABS(B)                                                    TABPRT55
      IF(B.LE.1.E-9 .OR. B.GE.9.E+18) GO TO 80                          TABPRT56
C                                                                       TABPRT57
C    REAL NUMBER                                                        TABPRT58
      FMT(II+1)=E5                                                      TABPRT59
      IF(B.LT.1.E-3 .OR. B.GE.1.E8) GO TO 90                            TABPRT60
   65 FMT(II+1)=F6                                                      TABPRT61
      IF(B.GE.1.E3) FMT(II+1)=F3                                        TABPRT62
      IF(B.GE.1.E5) FMT(II+1)=F1                                        TABPRT63
      GO TO 90                                                          TABPRT64
C                                                                       TABPRT65
C    INTEGER AND BCD                                                    TABPRT66
   80 FMT(II+1)=I12                                                     TABPRT67
      IF(IABS(IB).GT.IBCI) FMT(II+1)=BCD                                TABPRT68
      GO TO 90                                                          TABPRT69
C                                                                       TABPRT70
C    OCTAL                                                              TABPRT71
   85 FMT(II+1)=OCT                                                     TABPRT72
C                                                                       TABPRT73
   90 II    = II+1                                                      TABPRT74
      I     = I+1                                                       TABPRT75
      IF(I.GT.LSPACE) GO TO 100                                         TABPRT76
      IF(II.LE.NCOL .AND. I.LE.NB) GO TO 50                             TABPRT77
  100 I2    = I-1                                                       TABPRT78
      WRITE (6,FMT) I1,(A(I),I=I1,I2)                                   TABPRT79
      I1    = I2+1                                                      TABPRT80
  110 IF(I2.GE.NB) GO TO 990                                            TABPRT81
      IF(I.GT.LSPACE) GO TO 47                                          TABPRT82
      GO TO 48                                                          TABPRT83
  990 IITAB = 1                                                         TABPRT84
      RETURN                                                            TABPRT85
C                                                                       TABPRT86
 1000 FORMAT(/2X,A6)                                                    TABPRT87
 1002 FORMAT(1H )                                                       TABPRT88
      END                                                               TABPRT89
```

```
CTDSEQ         TRIDIAGONAL MATRIX-SIMULTANEOUS EQUATIONS
C
      SUBROUTINE TDSEQ(A,N2,ID,FN)
C*
C*    A=  COEFFICIENT ARRAY---A,B,C,D BY COLUMNS
C*  FIRST COLUMN DESTROYED BY SOLUTION VECTOR
C********       SOLUTION VECTOR RETURNED AS FIRST COLUMN OF A ARRAY
C*
C*
C*  IF NORMAL DERIVATIVE CONDITIONS ARE ENFORCED USING 1ST TWO
C*  POINTS ABOVE OR BELOW THE BOUNDARIES,  A(1) AND/OR A(3*N2).NE. 0.-
C*  THIS CONDITION IS FLAGGED BY IL AND/OR IU=1.
      DIMENSION A(800)
C*
      ENTRY TDSEQ1
      ISIZE=ID
      I2=ISIZE+1
      IF (A(I2)) 12,11,12
   11 FN=1.
      GO TO 50
   12 I3=2*ISIZE+1
      I4=3*ISIZE+1
      IU=0
      IL=0
      G=0.
      IF(A(1).NE.0.) IL=1
      INM=I3+N2-1
      IF(A(INM) .NE.0.) IU=1
      G=A(1)/A(I2)
      A(I3)=A(I3)/A(I2)
      A(1)=A(I4)/A(I2)
      N=N2
      DO 10 I=2,N
      I2=ISIZE+I
      I3=I2+ISIZE
      I4=I3+ISIZE
      IF(I.NE. N .OR. IU.NE.1) GO TO 15
      A(I)=A(I)-A(I3)*A(I3-2)
      A(I4)=A(I4)-A(I3)*A(I-2)
   15 CONTINUE
      A(I2)=A(I2)-A(I)*A(I3-1)
      IF (A(I2)) 13,11,13
   13 IF(I.EQ.2 .AND. IL.EQ.1) A(I3)=A(I3)-A(I)*G
      IF(I.EQ.N .AND. IU.EQ.1) GO TO 10
      A(I3)=A(I3)/A(I2)
   10 A(I)=(A(I4)-A(I)*A(I-1))/A(I2)
      I=N
   20 I=I-1
      IF (I) 40,40,30
   30 I3=2*ISIZE+I
```

```
      A(I)=A(I)-A(I3)*A(I+1)                                    TDSEQ051
      IF(IL.EQ.1 .AND. I.EQ.1) A(I)=A(I)-G*A(I+2)               TDSEQ052
      GO TO 20                                                  TDSEQ053
   40 FN=0.                                                     TDSEQ054
   50 RETURN                                                    TDSEQ055
      END                                                       TDSEQ056
```

```
CTFMH          CALCULATE TEMPERATURE FROM ENTHALPY
      SUBROUTINE TFMH (NK )
      LOGICAL  ENTRY1
      COMMON /INDATA/ N,HF,WAR,T2,BETA,T25,FARS,FINO2C,PO,
     *                RCO(11),RCO2(11),RHC(11),RNOX(11),
     *                PT(11),PS(11),BLOC(11),QCO(11)
      REAL            N
      COMMON /JETDAT/ NPTS,RAD(12),TS(12),UU(12),SPV(12),MWT(12),
     *                CP(12),FUUL(12),SPALDG(12),TKE(12),OTHER1(36)
      REAL            MWT
      COMMON /GASCMP/ RICH(12,2),FUEL(2,12,2),FNTH(2,12,2),
     *                CONC(16,2,12,2),HCINCP(12,2),OTHER(192)
      COMMON /GASTMW/ TG(2,12,2),MWTG(2,12,2),TAU(12,2),CPG(2,12,2)
      REAL            MWTG
      COMMON /OPCTRL/ TITLE(20),PRINT(30)
      COMMON /STCTRL/ LSTA,FINAL,CHEMK,FIRSTM,FIRSTC,XC,DXC,DUMST(10)
      LOGICAL         FINAL,FIRSTM,FIRSTC
      INTEGER         CHEMK
      COMMON /CPRINT/ PDUM(20)
      COMMON /GHSC  / GRT(25),HRT(25),SR(25),CPR(25),DCPR(25)
      DATA TG1,ENTRY1,R/3200.,.T.,1.98596/
C
C
      DO 100 K=1,NK
      DO 100 J=1,2
      HGT    = FNTH(J,K,2)
      IF( .NOT.ENTRY1 ) GO TO 10
      T1     = TG1
      GO TO 15
   10 T1     = TG(J,K,2)
   15 DO 30 KI=1,30
      CALL THRM( T1/1.8,1. )
      CPG1   = 0.
      HG1    = 0.
      DO 20 I=1,16
      CPG1   = CPG1+CONC(I,J,K,2)*CPR(I)
      HG1    = HG1+CONC(I,J,K,2)*HRT(I)
   20 CONTINUE
      HG1    = HG1*R*T1
      DH     = HGT-HG1
      CPG1   = CPG1*R
      DT     = DH/CPG1
      DTA    = ABS(DT)
      IF( DTA.LE. .01 ) GO TO 99
      T1     = T1+DT
      IF( T1.LE.189. ) T1=189.
   30 CONTINUE
      WRITE (6,35) T1,DT,HG1,DH
   35 FORMAT(//2X,19HTEMP. NOT CONVERGED/2X,4E16.8)
   99 TG(J,K,2)= T1
```

```
      CPG(J,K,2)= CPG1                                                  TFMH0051
  100 CONTINUE                                                          TFMH0052
      ENTRY1 = .FALSE.                                                  TFMH0053
C                                                                       TFMH0054
  200 RETURN                                                            TFMH0055
      END                                                               TFMH0056
```

```
CTFMH1        CP+ T FROM H
      SUBROUTINE TFMH1( C,H,T,CP )
      COMMON /GHSC  / GRT(25),HRT(25),SR(25),CPR(25),DCPR(25)
      DIMENSION C(1)
      LOGICAL  ENTRY1
      DATA TG1,ENTRY1,R/3200.,T,1.98596/
C
      HGT    = H
      IF( .NOT.ENTRY1 ) GO TO 1
      T1     = TG1
    1 DO 10 KI=1,30
      CALL THRM( T1/1.8,1. )
      CPG1   = 0.
      HG1    = 0.
      DO 5 I=1,16
      CPG1   = CPG1+C(I)*CPR(I)
      HG1    = HG1+C(I)*HRT(I)
    5 CONTINUE
      HG1    = HG1*R*T1
      DH     = HGT-HG1
      CPG1   = CPG1*R
      DT     = DH/CPG1
      IF( ABS(DT).LT. 0.01 ) GO TO 15
      T1     = T1+DT
      IF( T1.LE.189. ) T1=189.
   10 CONTINUE
      WRITE (6,14) T1,DT,HG1,DH
   14 FORMAT(//2X,19HTEMP. NOT CONVERGED/2X,4E16.8//)
   15 T      = T1
      CP     = CPG1
      ENTRY1 = .FALSE.
      RETURN
      END
```

```
CTHRUST        CALCULATE NET THRUST--LBF                                    THRUST01
      SUBROUTINE  THRUST(NX)                                                THRUST02
      INTEGER TWOJ, ITWO                                                    THRUST03
      LOGICAL SUPC,SUPSTP                                                   THRUST04
      LOGICAL SUBSON                                                        THRUST05
      LOGICAL TROUBL                                                        THRUST06
      LOGICAL DPRIN                                                         THRUST07
      LOGICAL LAST,CORSTP,ADDP,ENTRY1,IER                                   THRUST08
      LOGICAL  EOF  ,  ERR                                                  THRUST09
      LOGICAL  AXI , XPRN , CMPRS , QJET , TURBJ , CORE                     THRUST10
      REAL KCP,MUL,MUEFF,MACH                                               THRUST11
      REAL  MJET , ME , MUREF                                               THRUST12
      COMMON /JETTWO/                                                       THRUST13
     * TWO     , DIAO   , MJETO   , TJETO   , VJETO  ,                      THRUST14
     * PTJETO , TIJETO , NJO                                                THRUST15
      REAL  MJETO,MACHO                                                     THRUST16
      COMMON /3CO/  UO, EO, THO                                             THRUST17
      COMMON /CTRL2/                                                        THRUST18
     * EDGEI  , SFI    , MERGE  , XMERGE , YMERGE ,                         THRUST19
     * SLOPEI , SLOPEO , CEPTI  , CEPTO                                     THRUST20
      COMMON /MERGET/  MER, MERSTP , XMRG                                   THRUST21
      LOGICAL  TWO, MERGE , MER , MERSTP                                    THRUST22
      COMMON /SETNEW/  LEDGE, LCOREN                                        THRUST23
      COMMON /INP1  /         ENTRY1                                        THRUST24
      COMMON /MISC/ PM(10)                                                  THRUST25
      COMMON /PARAM/                                                        THRUST26
     * AL(200)  , BE(200)  , GM(200)  ,                                     THRUST27
     * EPS(200) , DL(200)  ,                                                THRUST28
     * VAR(200) , DVAR(200),                                                THRUST29
     * SM1(200) , NM1      , SM(200)  , NM        ,                         THRUST30
     * DX  ,                                                                THRUST31
     * B1  , C11 , D1  ,                                                    THRUST32
     * AN  , BN  , DN                                                       THRUST33
C*                                                                          THRUST34
C****** INPUT COMMON                                                        THRUST35
C*                                                                          THRUST36
      COMMON /INPJET/                                                       THRUST37
     * DIAJ       , MJET       , TJET       , PTJET      , VJET       ,     THRUST38
     * TIJET      ,                                                         THRUST39
     * PE         , VE         , ME         , TIE        , TE         ,     THRUST40
     * AXI        , NJ         , NMAX       ,                               THRUST41
     * XJ(100)    , XPRN(100)  ,                                            THRUST42
     * GAM        , RG         , PR         , PRT        ,                  THRUST43
     * SC         , TREF       , MUREF                                      THRUST44
C*                                                                          THRUST45
C****** CONTROL COMMON                                                      THRUST46
C*                                                                          THRUST47
      COMMON /CTRL/                                                         THRUST48
     * NXTA       , CMPRS      , QJET       , TURBJ      , COEF(10)   ,     THRUST49
     * NPU        , NPD        , DXC        , XU         , XDD        ,     THRUST50
```

```
     * DSTOR(800)
C*
C****** PROFILE COMMON
C*
      COMMON /PROF/ PSI(200),Y(200),UD(200),THD(200),ED(200)
C*
C****** CONSTANT AND ERROR COMMON
C*
      COMMON /CNFRR/ BITS , ERR , GC , GCJ , FOOT
C*
C****** BOUNDARY CONDITION COMMON
C*
      COMMON /BC/ UEDGE  , EEDGE  , THEDGE
C*
C****** POTENTIAL CORE COMMON
C*
      COMMON /CORED/ XCORE  , CORE  , CORSTP
C*
      COMMON /SUPER/ SUPC,SUPSTP,XSUP
C****** SCALER (UNITS CONVERSION) COMMON
C*
      COMMON /SCALER/  SP  , SV  , SLEN
C*
      COMMON /JET/
     * B(100)   , UC(100)  , TC(100)  , TIC(100) ,
     * PTC(100) , WJ(100)  , YJ(100)
C*
      COMMON /PROPJT/
     * P       , PRL      , PRTT     , RGAS     , SCC      ,
     * TREFF   , VSREF    , MACH     , XLC      ,
     * REFL    , C        , CHI      , RNORM    ,
     * RHO(200)   , MUL(200)    , KCP(200)    ,
     * MUEFF(200) , XLN(200)    , DK(200)     , RETURB(200)
      COMMON /XPRIN/ DPRIN
      COMMON /EDGE/  YJETE  ,  SFEDGE
      COMMON /UMFSH/ DUMU1(4),CXPC,CXTP,NRED
C*
      COMMON /MIXER/ MIX,RD(100),XD(100),CF,YR(100)
      LOGICAL  MIX
      COMMON /FLOBAL/ MAXIT,SUPB,NIT,PSID,YDD,YDC,
     *                 P1,P2,UCL,TOL,UPSTRM,CVG
      LOGICAL  SUPB,CVG,UPSTRM
      COMMON /ACONVG/ YCD(100),PD(100),INDC(100),  CHOKE, CHOKED
      LOGICAL  CHOKE, CHOKED
      COMMON /DFTT/  CLSP(100)
      COMMON /STA2/ MACH2,TS2,SS2,V2,RHO2,DPDX2
      REAL  MACH2
      COMMON /BCMIX2/ GRADU,TW,MUW,RHOW,PTE,TTE
      REAL  MUW
      COMMON /THERM/ GMC(200),CP(200)
```

```
C*
      COMMON /OUTMIX/ NXORIG
      COMMON /CBODY/ YCB(100),CLSPCB(100),YCB1,UCL1
      COMMON /THRST/  WV(100)
      DIMENSION STORI(200),THR(200)
      EQUIVALENCE  (STORI(1),AL(1)),(THR(1),BE(1))
      DATA  PI/3.14159265/
C*
C*  IF MIXING ZONE HAS INTERSECTED PLUG-- TERMINATE
C*
      EPS=1.
      IF(AXI)  EPS=2.
    1 YPLUG=YCB(NX)
      UP=YOF(YPLUG,Y,UD,1,NPD)
      IF(UP.EQ. UCL1)  GO TO 10
       IF(YPLUG .EQ. 0.)  GO TO 10
      WRITE  (6,100) NX
  100 FORMAT(//6X,28HMIXING ZONE INTERSECTED PLUG/,6X,7HSTATION,
     *  I7//12X,22HCALCULATION TERMINATED)
      ERR=.TRUE.
      GO TO 1000
C*
C*  CALCULATE THRUST OF JET STREAM
C*
   10 DO 15 L=1,NPD
      REXP=1.
      IF(AXI) REXP=Y(L)
   15 STORI(L)=RHO(L)*UD(L)*UD(L)*REXP
      THR(1)=0.
      CALL INTG(STORI,Y,THR,2,NPD)
      TJ=THR(NPD)/GC
C*
      IF(UPSTRM) GO TO 19
      TU=0.
      GO TO 50
C*
C*  THRUST OF UNENTRAINED FLOW
C*
   19 IF(NX.NE.1) GO TO 20
      TU=144.*PE/(RG*TE)*(VE/VJET)**2/(GC*EPS)*(YR(1)**EPS-Y(NPD)**EPS)
      GO TO 50
   20 TU=RHO2*(V2/VJET)**2/(GC*EPS)*(YDD**EPS-YJETE**EPS)
C*
C*
C*  CALCULATE NET THRUST
C*
   50 TERM=VJET**2*DIAJ/FOOT
      IF(AXI) TERM=PI*DIAJ*TERM/(2.*FOOT)
      TN=TERM*(TJ+TU)
C*
```

```
   60 WV(NX)=TN
C*
 1000 RETURN
      END
```

```
CTHRMM       CALCULATES (DIMENSIONLESS) THERMODYNAMIC PROPERTIES
C            FROM POLYNOMIAL CURVE FITS
      SUBROUTINE THRM (T,HONLY)
C
C     THIS ROUTINE CALCULATES (DIMENSIONLESS) THERMODYNAMIC PROPERTIES
C     FROM POLYNOMIAL CURVE FITS
C
      LOGICAL NEXT
C
      COMMON/COND/DUM(33),LS,LSP3,NEXT
      COMMON/GHSC/GRT(25),HRT(25),SR(25),CPR(25),DCPR(25)
      COMMON/TCOF/C(7,2,25),TLOW,TMID,THI
C
      F(T) = A1+T*(A2+T*(A3+T*(A4+T*A5)))
C
      IF (T .EQ. TPREV) RETURN
C
      IF (0.35*TLOW .LE. T  .AND.  T .LE. THI) GO TO 3
      IF (T .LE. 1.20*THI) GO TO 2
C
      WRITE (6,100) T
  100 FORMAT (7H0(THRM),5X,5HERROR,3X,3HT =,F8.2,16H IS OUT OF RANGE)
      NEXT = .TRUE.
      RETURN
C
    2 WRITE (6,101) T
  101 FORMAT (7H0(THRM),5X,7HWARNING,3X,3HT =,F8.2,16H IS OUT OF RANGE,
     * 4X,28HEXTRAPOLATED VALUES RETURNED)
C
C     LOCATE PROPER TEMPERATURE RANGE
    3 K = 2
      IF (T .GT. TMID) K = 1
C
      DO 4 I=1,LS
C
C     COMPUTE H/(R*T)
      A1 = C(1,K,I) + C(6,K,I)/T
      A2 = C(2,K,I)/2.
      A3 = C(3,K,I)/3.
      A4 = C(4,K,I)/4.
      A5 = C(5,K,I)/5.
    4 HRT(I) = F(T)
      IF (HONLY .EQ. 0.) RETURN
C
      TPREV = T
      DO 5 I=1,LS
C
C     COMPUTE G/(R*T)
      A1 = C(1,K,I)*(1.-ALOG(T)) + C(6,K,I)/T - C(7,K,I)
      A2 = -C(2,K,I)/2.
```

```
      A3 = -C(3,K,I)/6.                                                 THRMM051
      A4 = -C(4,K,I)/12.                                                THRMM052
      A5 = -C(5,K,I)/20.                                                THRMM053
      GRT(I) = F(T)                                                     THRMM054
C                                                                       THRMM055
C     COMPUTE S/R                                                       THRMM056
      A1 = C(1,K,I)*ALOG(T) + C(7,K,I)                                  THRMM057
      A2 = C(2,K,I)                                                     THRMM058
      A3 = C(3,K,I)/2.                                                  THRMM059
      A4 = C(4,K,I)/3.                                                  THRMM060
      A5 = C(5,K,I)/4.                                                  THRMM061
      SR(I) = F(T)                                                      THRMM062
C                                                                       THRMM063
C     COMPUTE CP/R                                                      THRMM064
      A1 = C(1,K,I)                                                     THRMM065
      A2 = C(2,K,I)                                                     THRMM066
      A3 = C(3,K,I)                                                     THRMM067
      A4 = C(4,K,I)                                                     THRMM068
      A5 = C(5,K,I)                                                     THRMM069
      CPR(I) = F(T)                                                     THRMM070
C                                                                       THRMM071
C     COMPUTE (DCP/DT)/R                                                THRMM072
      A1 = C(2,K,I)                                                     THRMM073
      A2 = 2.*C(3,K,I)                                                  THRMM074
      A3 = 3.*C(4,K,I)                                                  THRMM075
      A4 = 4.*C(5,K,I)                                                  THRMM076
      A5 = 0.                                                           THRMM077
    5 DCPR(I) = F(T)                                                    THRMM078
C                                                                       THRMM079
      RETURN                                                            THRMM080
      END                                                               THRMM081
```

```
CTRIDIA       SOLVES TRI-DIAGONAL SYSTEM OF EQUATIONS                    TRIDIA01
C                                                                        TRIDIA02
      SUBROUTINE TRIDIA (J,NN)                                           TRIDIA03
C                                                                        TRIDIA04
C*        THIS ROUTINE FINDS THE SOLUTION OF A TRIDIAGONAL SYSTEM OF EQUATRIDIA05
C* THE MATRIX -A- CONTAINS THE TRIDIAGONAL COEFFICIENT MATRIX            TRIDIA06
C* THE MATRIX -B- CONTAINS  THE RHS VECTOR                               TRIDIA07
C*  THE SIZE OF THE MATRIX IS (NN+1)X(NN+1)                              TRIDIA08
C                                                                        TRIDIA09
      COMMON /CTRIDI/ A(50,3),B(50)                                      TRIDIA10
      A(J,3)=A(J,3)/A(J,2)                                               TRIDIA11
      B(J)=B(J)/A(J,2)                                                   TRIDIA12
      JP1=J+1                                                            TRIDIA13
      NNM1=NN-1                                                          TRIDIA14
      DO 20 N=JP1,NNM1                                                   TRIDIA15
      A(N,2)=1./(A(N,2)-A(N,1)*A(N-1,3))                                 TRIDIA16
      A(N,3)=A(N,3)*A(N,2)                                               TRIDIA17
   20 B(N)=(B(N)-A(N,1)*B(N-1))*A(N,2)                                   TRIDIA18
C*        BACK SUBSTITUTION                                              TRIDIA19
C* STORE SOLUTION VECTOR IN RHS VECTOR LOCATION                          TRIDIA20
      N=NN                                                               TRIDIA21
      A(N,2)=1./(A(N,2)-A(N,1)*A(N-1,3))                                 TRIDIA22
      B(N)=(B(N)-A(N,1)*B(N-1))*A(N,2)                                   TRIDIA23
  260 IF (N-J) 300,300,270                                               TRIDIA24
  270 N=N-1                                                              TRIDIA25
      B(N)=B(N)-A(N,3)*B(N+1)                                            TRIDIA26
      GO TO 260                                                          TRIDIA27
  300 CONTINUE                                                           TRIDIA28
      RETURN                                                             TRIDIA29
      END                                                                TRIDIA30
```

```
CXSIZE          X-STEP SIZE CONTROL
      SUBROUTINE XSIZE (DX,X,REFL,NREG,LAST)
      LOGICAL LAST
      COMMON /SETNEW/  LKK,LC
      COMMON /UMESH/ DUMX(4),CXPC,CXTP,NRED
      COMMON /CTRL/ DUMSX(9), C6 , DUMSX1(8), XD  , DUMSX2(800)
C*
C*  NREG=1,  DX PROPORTIONAL TO B
C*  NREG=2,3 DX PROPORTIONAL TO JET RADIUS
C*
C*
      LAST=.FALSE.
      DXT=XD-X
    1 GO TO (10,20,20),NREG
C*
   10 DX=CXPC*REFL/C6
      IF(DX.GE.  .99*DXT)  DX=DXT
      XT=X+DX
      GO TO 50
C*
   20 IF(LC.GT.10)  GO TO 30
      DX=.1*FLOAT(LC)*CXTP*REFL/C6
      LC=LC+1
      GO TO 40
   30 DX=CXTP*REFL/C6
   40  IF(DX.GE.  .99*DXT)  DX=DXT
      XT=X+DX
C*
C*
C*  CHECK FOR X.GT. CALCULATION STATION
C*
   50 IF(XT.LT.XD)  GO TO 100
      LAST=.TRUE.
      DX=XD-X
C*
  100 RETURN
      END
```

```
CYOFX11                                                                 YOFX1101
      FUNCTION YOF(X1, X,Y,IA,IB)                                       YOFX1102
      COMMON /YOFXI/ I                                                  YOFX1103
      DIMENSION X(10),Y(10)                                             YOFX1104
      I1=IA+1                                                           YOFX1105
      I2=IB                                                             YOFX1106
      I=MIN0(MAX0(I1,I),I2)                                             YOFX1107
      N=I2                                                              YOFX1108
   40 IF(N) 42,999,42                                                   YOFX1109
   42 N=N-1                                                             YOFX1110
   53 F=(X1-X(I-1))/(X(I)-X(I-1))                                       YOFX1111
   55 IF(F) 60,100,70                                                   YOFX1112
   60 IF(I-I1) 65,100,65                                                YOFX1113
   65 I=I-1                                                             YOFX1114
      GO TO 40                                                          YOFX1115
   70 IF(F-1.) 100,100,72                                               YOFX1116
   72 IF(I-I2) 74,100,74                                                YOFX1117
   74 I=I+1                                                             YOFX1118
      GO TO 40                                                          YOFX1119
  999 CALL ERROR                                                        YOFX1120
      RETURN                                                            YOFX1121
  100 YOF =Y(I-1) + (Y(I)-Y(I-1))*F                                     YOFX1122
      RETURN                                                            YOFX1123
      END                                                               YOFX1124
```

SECTION 5.0

REFERENCES

1. Lyon, T.F., et. al., Development of Emissions Measurement Techniques for Afterburning Turbine Engines , AFAPL-TR-75-52, Air Force Aero Propulsion Laboratory, September 1975.

2. Heck, P.H., and Ferguson, D.R., Analytical Solution For Free Turbulent Mixing in Compressible Flows, AIAA Paper No. 71-4, January 1971.

3. Bittker, D.A., and Scullin, V.J., General Chemical Kinetics Computer Program For Static and Flow Reactions, With Application to Combustion and Shock-Tube Kinetics, NASA TN D-6586, January 1972.

4. Colley, W.C., Ferguson, D.R., Harsha, P.T., and Kenworthy, M.J., Scientific Technology of Hypersonic Ramjet Combustion Systems, AFAPL-TR-67-156, December 1967.

5. Spalding, D.B., "Concentration Fluctuations in a Round Turbulent Free Jet", Chemical Engineering Science, Volume 26, pp. 95-107, 1971.

6. Spalding, D.B., "Mathematical Models of Continuous Combustion", pp. 3-16 of Emissions from Continuous Combustion Systems, Cornelius, W. and Agnew, W.G. (Eds.), Plenum Press. 1972.

7. Lauder, B.E., and Spalding, D.C., Mathematical Models of Turbulence, Academic Press, 1972.

8. Anon., FORTRAN IV, Honeywell Information Systems, Pub. BN88.

ATE

LMED

- 7