CHEMICAL PROCESS SIMULATOR

WITH PLUG FLOW REACTOR

By

WILLIAM JOSEPH VEDDER JR.

Bachelor of Science in Chemical Engineering

Oklahoma State University

Stillwater, Oklahoma

1984

Submitted to the Faculty of the Graduate College
of the Oklahoma State University
in partial fulfillment of the requirements
for the Degree of
MASTER OF SCIENCE
December, 1985

CHEMICAL PROCESS SIMULATOR

WITH PLUG FLOW REACTOR

Thesis Approved:

Thesis Adviser

Dean of the Graduate College

PREFACE

This study has focused upon developing a general-purpose Chemical Reaction Engineering Simulator. The ideal plug flow reactor was chosen as the basis for this work.

I am sincerely appreciative of the outstanding support and technical assistance given me by my major adviser, Dr. Gary L. Foutch. I am also grateful to Dr. Mayis Seapan and Dr. Ruth C. Erbar for their technical assistance and continued interest throughout this work.

I thank Mrs. Daleene Caldwell for her professional work typing this manuscript. I especially thank my parents, Mr. and Mrs. William Vedder, for their support and encouragement throughout this endeavor.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The roll of computers in education has naturally evolved from rather limited usage in aiding scientific research to present applications in computer-assisted instruction (CAI) and state-of-the art process simulators. This broadening has mainly been due to the large decreases in computer costs and the desire to provide students with self-paced, individualized instruction without putting an exorbitant workload on the instructor. The self-paced method of instruction has been found to be a much more effective approach to teaching than the traditional lecture method (1).

Early uses of computers as aids in education include the work of Uttal (2) in 1962 at IBM who taught stenotyping through terminals connected to a computer system. Also in 1962, Lucklider (3) developed a system of math drills and graphical responses to be used in conjunction with a course in analytical geometry. Bitzer and Braunfeld started development of PLATO (Programmed Logic for Automated Teaching Operation) in 1967. PLATO is the oldest CAI system in existence (4). CAI grew rapidly in the 1970's and expanded to Europe and Japan (5, 6). Today research focuses on areas such as voice recognition systems and software that will better understand and process student responses. Much of the software currently being developed is in the form of new computer languages such as APL, originated by IBM; DECAL, authored by Digital

Equipment Corporation; and IPS, developed at Simon Fraser University (7).

The development of chemical process simulators began in the mid 1950's with the first published simulator appearing in 1958. Early historical reviews of process simulation are given by Mah (9) and Kehat and Shacham (10). Process simulators are not designed to replace an instructor as do the programs developed under the CAI system. Simulators are primarily tools used to produce better, cheaper designs quickly. At the university level, the use of simulation programs allows realistic, non-trivial design problems to be assigned which is important in keeping student interest. Particularly in the senior-level design course, the prudent use of a simulator should help the student translate his fundamental knowledge into a working, economically feasible design. The simulator, if used properly, allows the student to use his creative talents and quickly compare a number of different design cases.

Typical process simulators include the following broadly descriptive program modules:

1. Input data section - This section contains the process topology, design and unit operating parameters, and complete specifications of the process feed streams. The input data is checked by appropriate subroutines.

2. Data transfer section - The input data is transferred to the executive routine which is responsible for determining the order of program execution from the process topology data.

3. Unit calculation section - The individual process units are calculated here. If recycle streams are detected, a convergence routine is called to determine when stream data have converged. The executive

routine is responsible for program execution while the streams are converging.

4. Output data section - Here, the results of the simulation are made available to the user. These results may take the form of simple tables or graphics output.

In addition to the above sections, many newer simulators have routines that optimize a chemical process according to user supplied constraints (11).

The objective of this work was to develop a simulation program capable of modeling general kinetic schemes and calculating reactor volumes for an ideal, plug flow reactor (PFR). This program could then be used in conjunction with a course in Chemical Reaction Engineering to reinforce principles presented in the classroom. Questions could be answered such as:

1. Which types of reactions are best suited to be run in a PFR? For a given reaction scheme and conversion level, how large must the PFR be and how does this compare to the volumes of other reactors such as batch and continuous stirred tank?
2. How does recycle effect PFR volume? For a given reaction scheme, what is the optimum recycle ratio?
3. How does temperature and reaction order affect selectivity considerations for multiple reactions?

If experimental data are available relating concentrations and conversion levels, rate equations (represented by nonelementary power law kinetics) can be tested to determine the values of the exponents on the concentration terms.

CHAPTER II

LITERATURE REVIEW

Computers have become an increasingly important tool for use in Chemical Engineering education. Although they will not replace the formal lecture as the primary instructional mode, computers can and do serve as a useful adjunct to traditional courses in thermodynamics, reaction engineering, process design, process control and stagewise operation (12, 13).

Generally, three types of computer programs are in use today. The first type provides interactive instruction to students in the form of a series of lessons, each lesson being the equivalent of a homework-type assignment. The PLATO system developed at the University of Illinois, is the largest of the computerized teaching systems and is the only large system with full scale graphics capable of handling hundreds of terminals simultaneously (12). PLATO utilizes graphs, diagrams, animation, and projected slides and is capable of being programmed for highly complex problems. PLATO utilizes the TUTOR language, specifically developed to handle graphics capabilities and manipulate character strings. This language also allows new routines to be written by the instructors without a sophisticated knowledge of computer programming (7). No knowledge of computer programming is required of the students. Students need simply to complete an introductory lesson covering the use of a modified typewriter keyboard to fully utilize the

student mode of the PLATO system (14). This lesson is completed in 5 to 10 minutes.

The second type of program available is the process simulator. Simulators are mathematically derived models of actual industrial processes. Distillation towers, chemical reactors, flash separators, heat exchangers, compressors, and expanders are some of the units generally available as part of an overall process simulator. The user provides the simulation program with the desired flowsheet describing how the modules are connected, the feed composition, convergence criteria, and desired calculational procedures. The simulation's executive system supervises the calculation for each module and directs the communication between program segments. Early examples of process simulators are PACER, developed at Purdue University and introduced in the early 1960s, and CHESS, developed at the University of Houston and presented in 1968. FLOWTRAN, a proprietary program developed at Monsanto Company, was made available to universities in 1973 (12). Newer simulators include MAXI*SIM, an industrial simulator developed at Oklahoma State University and introduced in 1980. CHEMOS, written in PL/I to take advantage of the available pointer-addressed based data structures is a relatively new simulator developed at the University of British Columbia specifically for undergraduate students. It is a highly interactive process modeling program that includes multi-effect evaporators, feedback controllers, chemical reactors and many of the standard process modules such as splitters, mixers, and heat exchangers. A hand-calculator like language has also been built into the CHEMOS system which can be used in the interactive mode to perform special calculations. (15).

The proliferation of minicomputers in all areas of chemical engineering has spawned a need for miniaturized versions of the huge industrial simulators. The SIPRO-DTC flowsheeting simulation system, written in BASIC, has been successfully implemented on the desktop COMPUCORP 625 (equivalent of a Hewlett-Packard 35) and is powerful enough to solve "reasonable size" industrial problems (16).

The purpose of using process simulators in chemical engineering design courses is to help the student understand the relationships between the operating variables and the overall process design. Students can combine their creative input with the computational speed of the computer to reach a logical, economically justified design.

The third type of computer program available to students is the small (less than 500 lines) utility program. These programs cover specialized areas in simulation, data manipulation, and economic analysis. Utility programs are generally run in the batch mode with no student input during the simulation run.

A survey of the literature shows that while the tutorial program and process simulator are very powerful tools, the chemical reactor module is not represented as a design case, that is, the reactor volume is either known a priori or not even considered. In either case, input data will include extents of reaction or conversion and the operation mode (adiabatic or isothermal). The process simulator will then calculate the outlet stream. Tutorial programs may consider reactor types and optimum arrangements of reactors, however, once again, reactor volume is not calculated (17, 18).

An ideal program to be used in conjunction with a Chemical Reaction Engineering course should allow students to gain a further appreciation

of reactor design fundamentals. This could be accomplished by allowing the students to compare size requirements for different operating conditions or reactor configurations for a given production rate. Since a program designed to handle any type of reaction (elementary and nonelementary) in any type of reactor (ideal and nonideal) would be prohibitively large, the efforts in this study have focused on coming as close to the "ideal" program as possible given the time and resource constraints. This study has focused on using general reaction schemes (whose rate equations can be represented by elementary and nonelementary power law kinetics) but limiting the reactor configuration to gas phase, ideal plug flow. Isothermal or nonisothermal reactor operation can be specified. Recycle streams can be added by using the stream mixer and splitter modules. Thermodynamic data is available for the 61 most used components as specified by the Gas Processors Association. Output from the simulation includes a summary of the stream data, yield data, and data generated during the course of the integration. The program has been written so as to easily expand this reactor scheme to include other reactor types such as continuous stirred tank and batch reactors.

CHAPTER III

IDEAL PLUG FLOW REACTOR DESIGN PRINCIPLES

Tubular chemical reactors which exhibit, or are assumed to exhibit, concentration gradients only in the direction of flow are called plug flow reactors. Radial concentration gradients are assumed to be non-existent due to mixing action caused by the highly turbulent fluid flow.

One form of the design equation for a steady state plug flow reactor is

$$\frac{V}{F_{A_o}} = \int_{X_{A_i}}^{X_{A_f}} \frac{dX_A}{-r_A} \tag{3-1}$$

where: V = reactor volume, liters

F_{A_o} = molar flow rate of reactant A at reactor inlet

X_A = conversion of reactant A

$-r_A$ = rate of disappearance of reactant A, moles A/liter min

Subscripts i and f refer to conditions at the reactor inlet and outlet, respectively. Equation (3-1) is based on a material balance and is derived by several authors (19, 20, 21).

Equation (3-1) is completely general and used to calculate the volume of a plug flow reactor that will accomplish a desired conversion. The rate of disappearance of A, $-r_A$, is determined by the kinetics of the reaction and is temperature dependent as well as concentration dependent. For isothermal reactions possessing elementary

kinetics, Equation (3-1) can usually be integrated directly. However, integration of the design equation rapidly becomes unwieldy for cases involving multiple reactions, reactions possessing complex, nonelementary kinetics, and/or nonisothermal reactor operation. For these cases, numerical integration techniques must be used. Descriptions of these techniques can be found in the literature (21, 22, 23).

While these numerical techniques are suitable (though tedious) for hand calculations, many are not practical for computer calculations. This is because evaluation of the rate expression in terms of conversion and evaluation of the expansion factor becomes increasingly difficult for more than 2 reactions. It would be desirable to develop a general algorithm to be used for any sequence of reactions (both liquid and gas phase), thus eliminating the need for a complex series of subprograms which would handle only specific cases. This algorithm has been developed as follows:

1) Write the rate expressions, defining the rate of change of a component flow rate with respect to reactor volume, for each reaction in terms of a key component. The key component is defined as the first reactant in the stoichiometric equation.

2) The rate expressions for each component are then written as simple stoichiometric ratios of the rate expressions developed in (1).

3) The concentration terms in the rate equations are expressed as

$$C_i = \frac{F_i}{v} \tag{3-2}$$

where: C_i = concentration of the ith component, moles i/liter

v = total volumetric flow rate, liters/min

For the elementary reaction

$$2A + B \rightarrow 2C \tag{3-3}$$

the rate equations would be

$$-r_A = -\frac{dF_A}{dV} = k\frac{F_A^2 F_B}{v^3} \tag{3-4}$$

$$-r_B = -\frac{dF_B}{dV} = \frac{1}{2}k\frac{F_A^2 F_b}{v^3} \tag{3-5}$$

$$-r_C = -\frac{dF_C}{dV} = -k\frac{F_A^2 F_B}{v^3} \tag{3-6}$$

where: V = reactor volume, liters

k = reaction rate constant, liters^2/moles^2 min

The subscripts refer to particular components.

4) Equations (3-4) through (3-6) are integrated using the LSODAR (Livermore Solver for Ordinary Differential equations with Root finding) routine. For gas phase reactions, the component flows are summed at each point in the integration and a new volumetric flow rate calculated. This obviates the need for an expansion factor. Integration continues until the desired conversion is reached at which point values for flow rates, reactor volume, and temperature (if applicable) are returned.

Plug Flow Reactor Energy Balance

The energy balance for a steady-state plug flow reactor can be written

$$v \sum C_j C_{p_j} dT + \sum \Delta H_i r_i dV = -U dA_s (T-T^*) \qquad (3\text{-}7)$$

where: C_p = specific heat, cal/mol °C

T = temperature, °C

H = heat of reaction, cal/mol

U = overall heat transfer coefficient, j/s m^2 °C

A_s = surface area through which heat is transferred, m^2

T^* = wall temperature, °C

Subscripts j and i refer to the jth component and the ith reaction respectively. This equation assumes no energy generation other than by chemical reaction and no interphase transfer of heat. For a complete derivation of Equation (3-7) see reference (21).

For adiabatic reactor operation equation (3-7) can be rearranged as

$$\frac{dT}{dV} = \frac{-\sum \Delta H_i r_i}{v \sum C_j C_{p_j}} \qquad (3\text{-}8)$$

which is the form of the energy balance used in this program.

Plug Flow Reactor with Recycle

A simple recycle reactor is shown in Figure 1, adopted from Levenspiel (19).

The recycle ratio, R, is defined as

$$R = \frac{\text{volume of fluid returned to reactor entrance}}{\text{volume of fluid leaving the system}}$$

The design equation for a recycle reactor, applicable for any kinetics and any value of ϵ_a (expansion factor), can be written as

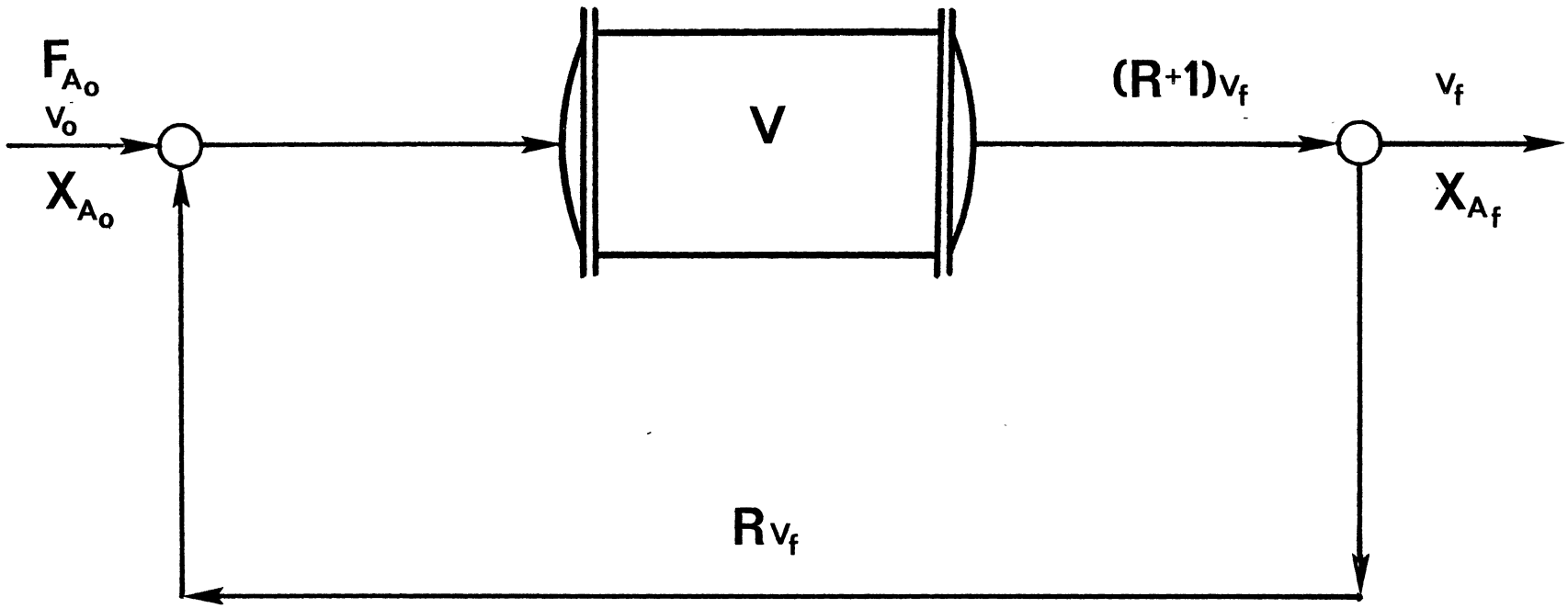

Figure 1. Typical Plug Flow Reactor with Recycle

$$\frac{V}{F_{A_o}} = (R + 1) \int_{(\frac{R}{R+1})X_{A_f}}^{X_{A_f}} \frac{dX_A}{-r_A} \qquad (3\text{-}9)$$

Aside from the problems covered earlier concerning integration of the design equation, Equation (3-9) cannot be used for this reactor simulation model because of the differences between the definitions of F_{A_o} and X_A used in Equation (3-9) and F_{A_o} and X_A used in the simulator. F_{A_o} used in Figure 1 and defined in Equation (3-9) is based on the flow of A to the process where F_{A_o} used in the simulation is taken as the flow of A at the reactor inlet. As the simulator iterates to a converged solution, the flow of A starts at zero and ends at some final value. F_{A_o} is continually changing and thus is not known a priori. The conversion of component A, X_A, requested by the program and calculated based upon F_{A_o}, is the single-pass conversion and not the overall conversion, X_{A_f}, in Equation (3-9). Thus, X_{A_f} is also an unknown.

Therefore, recycle reactor volumes are calculated using the method of successive substitution. Recycle stream flows are initially set equal to zero, then updated during the iteration process. Convergence is obtained when the difference in all component molar flow rates between two successive iterations is less than .01%.

CHAPTER IV

PROGRAM DESCRIPTION

General Program Capabilities

The main purpose of this research was to develop an interactive computer program capable of simulating a chemical reactor system and calculating reactor size requirements for the steady-state, ideal plug flow case. KINETX, the name of the program developed in this study, allows the user to see the effects that reactor operating conditions (e.g. pressure, isothermal or nonisothermal operation) and process stream arrangements (such as recycle streams) have upon reactor volumes and product yields. KINETX can simulate a gas phase process consisting of a maximum of five units (any combination of mixers, splitters, and plug flow reactors) and nine streams. The maximum number of components allowed in the process is nine. Up to five chemical reactions can be simulated simultaneously. There are no constraints on the type of reactions (elementary, nonelementary, series, parallel), however, the stoichiometric coefficients must be integers between one and nine and there must be no more than five components per reaction. KINETX is written in FORTRAN 77 and implemented on an IBM 3081K mainframe computer.

Constraints on Unit Modules

Each unit module has constraints that limit the number of feed and

product streams associated with the unit and the unit's mode of operation. The mixer module allows up to four inlet streams and will calculate one outlet stream. Isothermal or nonisothermal operation can be specified. The splitter module will split one inlet stream into a maximum of four outlet streams and operates isothermally only. The plug flow reactor (PFR) module will accept one inlet and one outlet stream only. Either isothermal or nonisothermal operation can be specified.

Program Structure

KINETX sequentially executes the following steps:

1. User identification and title are entered. Global variables are declared, dimensioned, and initialized.

2. Process topology (describing how the streams and units are arranged) is requested. The user selects the units to be simulated and enters the stream numbers for the streams associated with each unit. Once the topology has been set, it is checked by the program for violation of any topological constraints. If any violations are found, an appropriate error message is issued and the simulation is restarted.

3. The process and stream connection matrices are filled with the appropriate unit numbers, unit identification numbers, and stream numbers. A full description of these matrices is given by Crowe et. al. (24). Streams are flagged as process feed streams, process effluent streams, or streams connecting two units.

4. Stream data are requested for process feed streams. All data are checked to make sure they fall within acceptable limits.

5. The reaction type and stoichiometric equations are now entered. From this input data, a coefficient matrix is set up. Again,

all input data are checked for acceptability.

6. Any parameters characteristic of a unit module are requested. This completes the input of data. The user may then choose to view the simulation prior to execution and make any necessary corrections.

7. The simulation is ready to be executed. Unit modules are calculated in the same sequence as they were entered.

8. Results of the simulation can be displayed for three different sets of data. The first set is stream information and consists of component flow rates (kg-mol/min and kg/min), mass and mole fractions, and enthalpies and temperatures (if applicable). Yield data, relative to reactant "A" is presented in the second data set. The third data set is generated during the integration and consists of component flow rates, temperatures, and reactor volume as a function of conversion.

Program Organization and Subroutine Description

Figure 2 shows the overall program organization. A short description of the program subroutines follows.

INISHL

The function of INISHL is to initialize global variables. This routine is called at the beginning of each simulation run.

TOPOLI

Subroutine TOPOLI requests data concerning the topology of the desired simulation. These data are: 1) Number and types of units to be simulated and 2) stream I.D. numbers for the streams associated with each unit. The stream I.D. numbers are read as character data (as

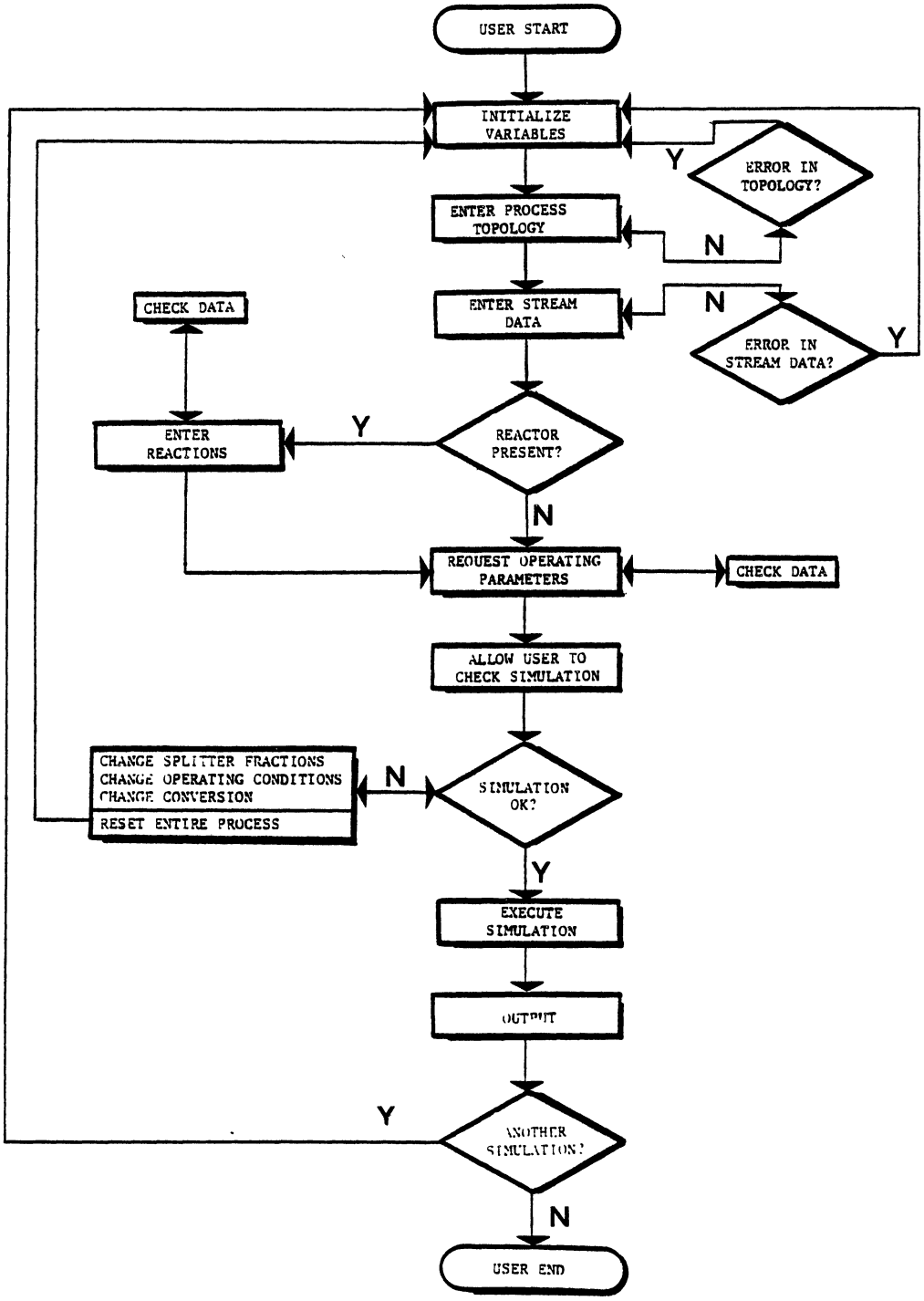

Figure 2. Program Organization

opposed to numeric data), which are compared to an array holding all acceptable characters. When a match between the input character and array character is found, the position of the character in the array is used to set the stream number. From this data, the process matrix is set. Each row in the process matrix corresponds to a process unit and holds the unit number, (which determines the order in which the unit is calculated), the unit I.D. number, feed streams to the unit, and product streams from the unit. TOPOLI uses calls to the routines IERRLM and IOCHK to check that limits on integer input data and constraints on the number of inlet and outlet streams associated with each unit, have been met.

STNSCM

Subroutine STNSCM sets the stream connection matrix (NSCM) given the information in the process matrix (NPM), as described earlier.

YIELD

Subroutine YIELD calculates and displays the yields of user specified components. Both the fractional yield based on feed and yield based on the amount of the key component that reacts are calculated.

STRMI

The function of STRMI is to request and accept process feed stream data, such as component I. D. numbers, flow rates, initial concentrations, and temperature and pressures (if needed). All stream data are checked through calls to IERRLM and RERRLM prior to loading the data into the STRMID array.

IERRLM and RERRLM

These routines check integer and real input data, respectively, to make sure the data fall within acceptable limits. If an error is found in the data, an appropriate error message is printed, the request for data is repeated, and the new value is checked.

RXNI

Subroutine RXNI accepts the types of reactions to be modeled (elementary or nonelementary) and the stoichiometric equations. Since the equations are read as character arrays by the program, the following sequential format for input of each reaction is used.

a. The first character in each equation must be numeric between one and nine.

b. The second character must be alphabetic.

c. The third character must be a separator (+, >, or =). A '>' represents an irreversible reaction while the '=' signifies a reversible reaction.

If this sequence is not followed, an appropriate error message is displayed and a request to reenter the data is printed. Subroutine RXNI also sets up the coefficient matrix, NCOEF, which is used to determine the reaction rate equations for each component. The functions of the coefficient matrix are further discussed by Henley and Rosen (25).

MXMNIO

SUBROUTINE MXMNIO searches the stream connection matrix and issues an error message if there are more than six process feed or product streams. Calls are made to INISHL, TOPOLI, and STNSCM to reset the

topology if the error message is displayed.

STUNIT

Subroutine STUNIT reads the process matrix and makes appropriate requests for unit operating parameters. These parameters include isothermal/nonisothermal operation and splitter product stream fractions.

EXEC

Subroutine EXEC handles the execution of the unit modules. EXEC reads the second column of the process matrix, which holds the unit I.D. numbers, and issues calls to the appropriate subroutine module. For recycle calculations, EXEC continues to make calls to the modules until all the streams have converged.

MIXER

The function of the MIXER subroutine is to simulate an adiabatic mixer. The argument for this subroutine, INDX, is the unit number from the first column of the process matrix. Thus, the argument distinguishes mixer modules if more than one mixer exists in the process.

MIXER issues calls to NOUTØ(NIN), NOUTØ(NOUT), FNDOUT(INDX, NOUT, NUMOUT), and FNDIN(INDX, NIN, NUMIN) prior to the calculation of the outlet stream. The first two calls initialize the NOUT and NIN arrays to zero. The last two calls initiate a search of the process matrix to determine the product and feed streams associated with the unit. NOUT and NIN are sequentially filled with the product and feed stream I.D. numbers respectively. NUMOUT and NUMIN are the respective number of outlet and inlet streams linked to the unit.

If the unit is to run nonisothermally, STTEMP is called to set the temperatures of the process feed streams associated with the unit. If recycle streams are present, the convergence flag for each unit, NFLGCV, is set equal to one.

The actual mixing calculations are performed in subroutine MIX. After the outlet stream flows have been loaded into the STRMID array, TFACE is called to interface with the thermodynamic property data bank (26) and calculate mass flows, mass and mole fractions, and enthalpies, if needed. If a mixer is running nonisothermally, TFACE issues a call to FNDTMP which calculates the temperature of the outlet stream. The outlet temperature is iterated using the Newton-Raphson technique, until the energy balance is satisfied.

SPLTTR

Subroutine SPLTTR simulates an adiabatic, isothermal stream splitter. Calls to NOUTØ, FNDOUT and FNDIN set up the splitter calculation. If needed, subroutine STOUTT is called to set the temperatures of the outlet streams. Subroutine SPPRMS is called from STUNIT and requests the user to specify the fraction of the inlet stream that is to flow to an outlet stream. This request is made N-1 times where N is the number of outlet streams. Subroutine SPLT uses the splitter parameters from SPPRMS, and performs the calculations that set the outlet stream molar flows.

PFR

Subroutine PFR simulates an adiabatic plug flow reactor. Only gas phase reactions can be specified. Calls to subroutines NOUTØ, FNDOUT,

and FNDIN set up the PFR calculation. For gas phase reactions, subroutine FNDNRT will search the feed stream for inerts and set the variable FLONRT equal to the sum of molar flows of all the inerts. Subroutine STX requests the conversion of the reactant. For recycle reactors, this conversion is the single-pass conversion. If elementary reactions have been specified, subroutine STEXP is called from STUNIT to convert stoichiometric coefficients in NCOEF to real numbers and store the numbers in the EXP array. For nonisothermal reactor operation, calls from STUNIT are made to STE, STA, and STDELH. These calls request the activation energy, the Arrhenius constant, and the heat of reaction for each reaction, otherwise STRK is called to request rate and equilibrium constants.

STPOS

The function of STPOS is to sequentially fill the array NPOSPC with the positions (i.e. row numbers) of the reactants and products in the NCOEF array. Subroutine FEX uses the NPOSPC array to distinguish which species are to be included in a component rate equation.

The final step before set-up and integration of the rate equations is a call to TFACE which handles the calculations using the thermodynamic property data bank.

ODE

Subroutine ODE's main functions are to:

1) Set the parameters required by the integration routine, LSODAR. These parameters include error tolerances, operating flags, and array dimensions.

2) Load the Y array with the inlet component flow rates. In effect, these are the initial conditions of the differential equations to be solved.

3) Call LSODAR for each point during the integration at which an answer is desired.

4) Issue error messages if the integration procedure is unsuccessful.

5) Transfer the results of the integration to the outlet stream array.

FEX

Subroutine FEX is the heart of the PFR module. FEX reads the appropriate arrays then sets up and evaluates the rate equation for each component participating in the reaction(s). For simplicity, a rate equation for each reaction based on a key component is set-up. The component rate equations are then stoichiometric ratios of the rate equation for the key component. For example, if the reactions are:

$$2A + B \longrightarrow 2C \qquad (4\text{-}1)$$

$$B + C \longrightarrow 2D \qquad (4\text{-}2)$$

then, using A as the key component in (4-1) and B as the key component in (4-2), the rate equations are (assuming elementary kinetics):

$$(-r_A)_1 = -\left(\frac{dF_A}{dV}\right)_1 = k_1 \frac{F_A^2 \; F_B}{v^3} \qquad (4\text{-}3)$$

$$(-r_B)_2 = -\left(\frac{dF_B}{dV}\right)_2 = k_2 \frac{F_B F_C}{v^2} \qquad (4\text{-}4)$$

The overall component rates are then stoichiometric ratios of the key components' reaction rates. Thus the overall component rates of disappearance are:

$$-r_A = -\frac{dF_A}{dV} = \left(\frac{-2}{-2}\right) k_1 \frac{F_A^2 F_B}{v^3} \tag{4-5}$$

$$-r_B = -\frac{dF_B}{dV} = \left(\frac{-1}{-2}\right) k_1 \frac{F_A^2 F_B}{v^3} + \left(\frac{-1}{-1}\right) k_2 \frac{F_B F_C}{v^2} \tag{4-6}$$

$$r_C = \frac{dF_C}{dV} = \left(\frac{2}{-2}\right) k_1 \frac{F_A^2 F_B}{v^3} - \left(\frac{-1}{-1}\right) k_2 \frac{F_B F_C}{v^2} \tag{4-7}$$

$$r_D = \frac{dF_D}{dV} = \left(\frac{2}{-1}\right) k_2 \frac{F_B F_C}{v^2} \tag{4-8}$$

Equations (4-5) to (4-8) can now be integrated until the desired conversion of A is reached. Subroutine GEX supplies the constraint function, GOUT, that is driven to zero as the desired conversion is reached. This function is

$$\text{GOUT}(1) = 1. - \frac{F_A}{F_{A_o}} - X_D \tag{4-9}$$

where: X_D = desired conversion of A

Thus, LSODAR integrates the rate equations and it evaluates GOUT(1) to determine where to stop the integration. When GOUT(1) reaches zero, the integration stops and the current values for the flow rates and independent variable (reactor volume) are saved in the RCT array.

LSODAR

Subroutine LSODAR is a variant version of LSODE written by Alan C.

Hindmarsh and Linda R. Petzold. LSODAR solves systems of first order, ordinary differential equations (ODEs) of the form;

$$\frac{dy(i)}{dt} = f(i, t, y(1), \ldots, y(NEQ)) \qquad (4\text{-}10)$$

$$i = 1, \ldots, NEQ$$

where: y = dependent variable, (flow rate or concentration)

t = independent variable, (reactor volume)

NEQ = number of equations.

LSODAR distinguishes between stiff and nonstiff problems and automatically implements the appropriate method of solution (27). In addition to solving the set of rate equations, LSODAR finds the root of Equations (3-9) and returns the values for the component flow rates (or concentrations), reactor volume, and outlet temperature (for nonisothermal reactor operation) at that root.

The algorithm used by LSODAR is based on the Adams-Moulton (A-M) multistep method for solving ODEs (28). Multistep methods utilize past values of y and/or y' to construct a polynomial that approximates y' over an interval where the solution has already been calculated. The polynomial is then extrapolated to the next interval where a solution is desired. Better approximate solutions are obtained until the difference between two successive approximations is sufficiently small. This "built-in" accuracy criteria allows step sizes to be easily adjusted when local errors are less than some user supplied tolerance (28, 29).

CHAPTER V

PROGRAM TESTING AND RESULTS

The program developed in this research was tested for accuracy and reliability by comparing problem solutions available in the literature and other sources to solutions generated by the program. Because the input data for many of the reactor design problems taken from the literature did not conform to what was required by the program, the literature problem had to be restructured. This usually occurred when space-times instead of reactor volumes were calculated. For this case, the restructuring was minor. Molar flow rates into the reactor were assumed to correspond to the feed mole fractions given in the literature problem. Using the temperature and pressure data given in the problem statement, an inlet volumetric flow rate is calculated. The reactor volume can then be calculated from the volumetric flow rate and space-time. Other restructuring may be more complicated such as extracting the Arrhenius constant and activation energy from rate constant/temperature data.

Testing and Evaluation of LSODAR

The LSODAR integration routine was tested by running an example problem that was provided with the routine. Since LSODAR was tested extensively at the Lawrence Livermore Laboratory prior to its release, the main objective of this test was to determine if any errors were

introduced into the routine during transfer from the VAX 11/780 system to the IBM 3081D.

This example problem is taken from chemical kinetics:

$$A \underset{k_2}{\overset{k_1}{\rightleftharpoons}} B + C \tag{5-1}$$

$$2B \xrightarrow{k_3} C \tag{5-2}$$

where: $k_1 = .04$

$k_2 = 10^4$

$k_3 = 3 \times 10^7$

A mass balance in a batch reactor yields the following rate expressions:

$$\frac{dy_1}{dt} = -.04y_1 + 10^4 y_2 y_3 \tag{5-3}$$

$$\frac{dy_2}{dt} = .04y_1 - 10^4 y_2 y_3 - 3x10^7\ y_2^2 \tag{5-4}$$

$$\frac{dy_3}{dt} = 3x10^7\ y_2^2 \tag{5-5}$$

where y_1, y_2, and y_3 represent the concentrations of components A, B, and C respectively. The initial conditions are given by

$$y_1(0) = 1.0 \tag{5-6}$$

$$y_2(0) = y_3(0) = 0.0 \tag{5-7}$$

Equations (5-3) through (5-5) are considered highly stiff and their solutions present a severe test to any ODE solver (30, 31, 32, 33). No coding of these equations was necessary. Activation of the routine was done by removing the appropriate comment cards.

The results of the LSODAR evaluation are shown in Table I. A Control Data Corporation 7600 Series computer (in single precision) was used to generate the output that was supplied with the LSODAR program (27). Small discrepancies between the solutions computed on the machines start to occur at t = 4.0000E+07. However, these discrepancies arise from differences in the way the IBM and CDC computers perform their floating-point operations and have not been caused by coding errors. White and Seider (34) tested LSODE (which uses the same integration procedure as LSODAR) on large, complex combustion reaction systems. They found that numerical instability occurs as the result of using inefficient absolute error tolerances when evaluating the algebraic mass balances which account for the initiation of free radicals. However, the reactions to be modeled for this work will be relatively simple systems and no integration problems should arise.

Testing and Evaluation of the Simulator Using the Mixer and Splitter Modules Only

During the course of this program's development, many tests were made on the mixer and splitter modules. The tests consisted of simulating different arrangements of the units and checking the output against mole balances done around each unit and the entire process. Presented here is a test that incorporates the maximum number of components and maximum number of streams allowed for any simulation. A

TABLE I

RESULTS OF THE LSODAR INTEGRATION OF EQUATIONS (5-3) - (5-5)

	CDC-7600			IBM 3081D		
t	y_1	y_2	y_3	y_1	y_2	y_3
4.0000E-01	9.851712E-01	3.386380E-05	1.479493E-02	9.85171D-01	3.38638D-05	1.47949D-02
4.0000E-00	9.055333E-01	2.240655E-05	9.444430E-02	9.05533D-01	2.24066D-05	9.44443D-02
4.0000E-01	7.158403E-01	9.186334E-06	2.841505E-01	7.15840D-01	9.18633D-06	2.84150D-01
4.0000E-02	4.505250E-01	3.222964E-06	5.494717E-01	4.50525D-01	3.22296D-06	5.49472D-01
4.0000E-03	1.831975E-01	8.941774E-07	8.168016E-01	1.83198D-01	8.94177D-07	8.16802D-01
4.0000E-04	3.898730E-02	1.621940E-07	9.610125E-01	3.89873D-02	1.62194D-07	9.61023D-01
4.0000E-05	4.936363E-03	1.984221E-08	9.950636E-01	4.93636D-03	1.98422D-08	9.95064D-01
4.0000E-06	5.161831E-04	2.065786E-09	9.994838E-01	5.16183D-04	2.06579D-09	9.99484D-01
4.0000E-07	5.179817E-05	2.072032E-10	9.999482E-01	5.17981D-05	2.07203D-10	9.99948D-01
4.0000E-08	5.283401E-06	2.113371E-11	9.999947E-01	5.28362D-06	2.11346D-11	9.99995D-01
4.0000E-09	4.659031E-07	1.863613E-12	9.999995E-01	4.65876D-07	1.86350D-12	1.00000D-01
4.0000E-10	1.404280E-08	5.717126E-14	1.000000E+00	1.42854D-08	5.71416D-14	1.00000D-01

successful test should then indicate that the program is reliable for a simulation of any non-reactive process.

The process to be simulated is shown in Figure 3. Results of the simulation from this work and the MAXI*SIM process simulator are shown in Table II. All values are essentially in complete agreement. The minor differences that do exist result from different convergence criteria used by the simulators. MAXI*SIM tests all stream data (flow rates, temperatures, pressures, enthalpies, etc.) for convergence while the simulator developed in this work tests only component flow rates. Since flow rates converge faster than the other stream properties MAXI*SIM will continue to perform calculations on the component flow rates even though these variables have met the convergence tolerance. This effectively tightens the error criteria for the flow rates. As a result of this test case, a "user supplied" recycle tolerance was incorporated into this work.

Demonstration Problems: Comparison of Simulator Results with Literature Values for Reacting Systems

General Background

Four example problems will be presented here. These have been selected to test the accuracy of the simulator and demonstrate the wide range of problems that can be solved by the program. An important point to keep in mind is that there are no subprograms that solve (either numerically or analytically) specific PFR design equations from a preprogrammed set of rate equations. Each problem is unique in that subroutine FEX evaluates the appropriate arrays to determine the form of the component rate equations to be solved. So each test case is

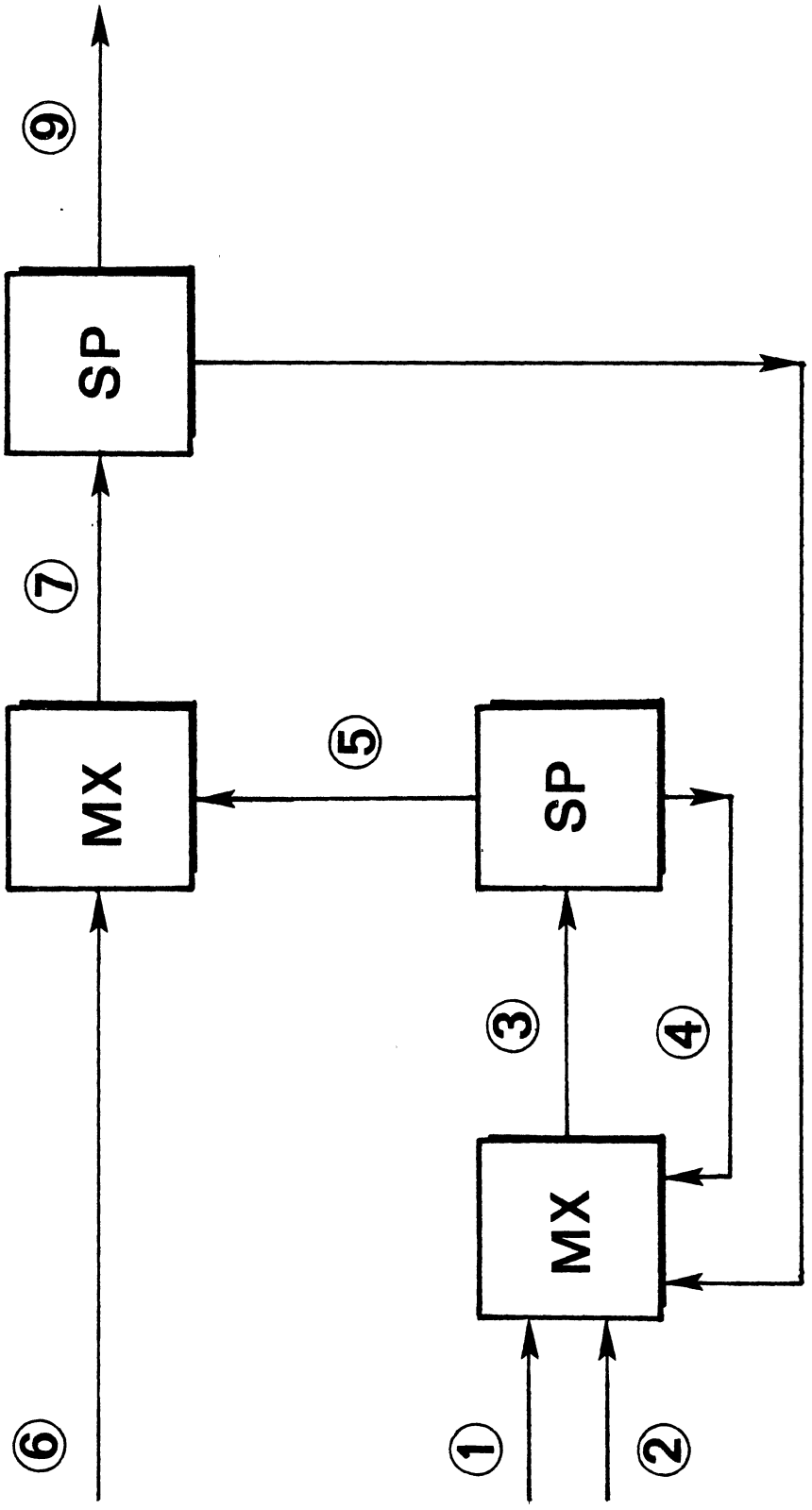

Figure 3. Process Flow Diagram of the Mixer/Splitter Simulation

TABLE II

COMPARISON OF THE RESULTS FOR PROCESS
SIMULATION CASE 2

	Stream 1[a]		Stream 2[a]	
COMPONENT	MAXI*SIM	THIS WORK	MAXI*SIM	THIS WORK
H2	100.0000	100.0000	0.0000	0.0000
CH4	200.0000	200.0000	0.0000	0.0000
C2H6	100.0000	100.0000	0.0000	0.0000
C3H8	100.0000	100.0000	400.0000	400.0000
IC4H10	70.0000	70.0000	300.0000	300.0000
NC4H10	50.0000	50.0000	0.0000	0.0000
IC5H12	30.0000	30.0000	0.0000	0.0000
NC5H12	20.0000	20.0000	0.0000	0.0000
NEO-C5	10.0000	10.0000	0.0000	0.0000
TOTAL	680.0000	680.0000	700.0000	700.0000
T (°C)	800.00	800.00	100.00	100.00
H (MCAL)	15512.53	15511.78	3844.05	3857.83

[a]Process feed stream.

NOTE: All stream flows are in Kg·mols

TABLE II (Continued)

	Stream 3		Stream 4	
COMPONENT	MAXI*SIM	THIS WORK	MAXI*SIM	THIS WORK
H2	333.3334	333.3286	100.0000	99.9986
CH4	666.6667	666.6572	200.0000	199.9972
C2H6	333.3334	333.3286	100.0000	99.9986
C3H8	1285.7145	1285.6980	385.7144	385.7095
IC4H10	947.6191	947.6077	284.2857	284.2822
NC4H10	166.6667	166.6644	50.0000	49.9993
IC5H12	100.0000	99.9987	30.0000	29.9996
NC5H12	66.6667	66.6658	20.0000	19.9997
NEO-C5	33.3333	33.3329	10.0000	9.9999
TOTAL	3933.3335	3933.2819	1180.0001	1179.9800
T (°C)	395.05	395.3306	395.05	395.3306
H (MCAL)	48914.00	48963.79	14674.18	14689.1310

TABLE II (Continued)

COMPONENT	Stream 5		Stream 6[a]	
	MAXI*SIM	THIS WORK	MAXI*SIM	THIS WORK
H2	233.3333	233.3300	100.0000	100.0000
CH4	466.6667	466.6599	200.0000	200.0000
C2H6	233.3333	233.3300	100.0000	100.0000
C3H8	900.0001	899.9893	100.0000	100.0000
IC4H10	663.3333	663.3252	70.0000	70.0000
NC4H10	116.6667	116.6651	50.0000	50.0000
IC4H12	70.0000	69.9991	30.0000	30.0000
NC5H12	46.6667	46.6660	20.0000	20.0000
NEO-C5	23.3333	23.3330	10.0000	10.0000
TOTAL	2753.3333	2753.3000	680.0000	680.0000
T (°C)	395.05	395.33	100.00	100.00
H (MCAL)	34239.75	34274.65	2981.51	2987.80

[a]Process feed stream

TABLE II (Continued)

	Stream 7		Stream 8	
COMPONENT	MAXI*SIM	THIS WORK	MAXI*SIM	THIS WORK
H2	333.3333	333.3298	133.3333	133.3319
CH4	666.6667	666.6599	266.6667	266.6638
C2H6	333.3333	333.3298	133.3333	133.3319
C3H8	1000.0001	999.9893	400.0001	399.9956
IC4H10	733.3331	733.3252	293.3333	293.3298
NC4H10	166.6667	166.6651	66.6667	66.6660
IC5612	100.0000	99.9991	40.0000	39.9996
NC5H12	66.6667	66.6660	26.6667	26.6664
NEO-C5	33.3333	33.3330	133.3333	13.3332
TOTAL	3433.3335	3433.30	1373.3334	1373.32
T (°C)	351.53	351.91	351.42	351.91
H (MCAL)	37210.76	37261.46	14880.12	14904.97

TABLE II (Continued)

	Stream 9	
COMPONENT	MAXI*SIM	THIS WORK
H2	200.0000	199.9979
CH4	400.0000	399.9958
C2H6	200.0000	199.9979
C3H8	600.0001	599.9934
IC4H10	440.0000	439.9951
NC4H10	100.0000	99.9991
IC5H12	60.0000	59.9994
NC5H12	40.0000	39.9996
NEO-C5	20.0000	19.9998
TOTAL	2060.0002	2059.9780
T (°C)	351.42	351.91
H (MCAL)	22320.19	22357.46

primarily an evaluation of FEX's ability to develop the rate expressions which correctly describe the kinetics of the reactions and to a lesser extent the ability of the executive system to direct program execution. It is also important to realize that only for non-isothermal reactions (where specific heat data are used), is it important to specify the reactive species by their component ID numbers. For isothermal cases which involve hypothetically reacting components such as 'A' or 'R', specifying a component from the data bank will have no effect upon the volume calculation; any component ID number may be entered.

Demonstration Problems

Test Case I.

This example, taken from Holland (20) page 59, is the thermal, gas phase reaction of methane with sulfur at 600 °C.

$$CH_4 + 2S_2 \longrightarrow CS_2 + 2H_2S \tag{5-8}$$

The rate of disappearance of sulfur is given by

$$-r_{S_2} = k_c \, C_{CH_4} \, C_{S_2} \tag{5-9}$$

Total pressure is one atmosphere. The feed rate of methane is 23.8 moles/hr and that of sulfur is 47.6 moles/hr. The problem requires calculation of the residence time for an 18% conversion of methane. Note that the reaction is nonelementary.

Other than converting the rate constant and molar flows into the

units required by the simulator, this problem requires no restructuring. Holland calculates a volumetric flow of 5.115×10^6 ml/hr and a space time of .003938 hr. From this, the volume is calculated as 20.14 liters. The result from the simulator is 20.15 liters which fully agrees with Holland.

Test Case II.

This example is taken from Hill (35), page 362, and involves the nonisothermal, gas phase reaction of butadiene with ethylene.

$$C_4H_6 + C_2H_4 \longrightarrow C_6H_{10} \qquad (5\text{-}10)$$

The problem states that an equimolar mixture of butadiene and ethylene is fed to a plug flow reactor at 723 K. The space time for a 10% conversion of butadiene is desired. Hill calculates a space time of 47.11 seconds and an outlet temperature of 775 K. Using this data and assuming an initial feed of 20 kg mol/min for each reactant, the volume is calculated as 186,400 liters. The results from the simulation show a volume of 182,600 liters and an outlet temperature of 777 K. This difference can be attributed to the absence of specific heat data for cyclohexene (data was used for cyclohexane) and the assumption of constant specific heats used by Hill whereas the simulator accounts for the temperature dependency of the specific heats. The values of the specific heats used by Hill are 20.2, 36.8 and 59.5 cal/mol K for ethylene, butadiene, and cyclohexene, respectively. The program calculates a range of specific heats from 19.02, 35.21, and 62.39 cal/mol K at 723 K to 19.77, 36.48, 65.64 cal/mol k at 777 K. Assuming

any effect of the higher cyclohexane specific heat is negligible (due to the low conversion), a higher outlet temperature is to be expected. This higher temperature would also drive the reaction faster, thus accounting for the smaller reactor volume calculated by the simulator.

Test Case III.

This next example is provided by Holland (20), page 67, in the form of an integral derived for the following reversible reaction.

$$A + B \underset{k_p}{\overset{k_p}{\rightleftharpoons}} R \qquad (5\text{-}11)$$

Holland's solution is

$$\frac{k_p\, y_{A_o}\, PV}{K_p\, N_{A_o}} = \int_0^x \frac{(1-y_{A_o} X)^2 dX}{aX^2 + bX + C} \qquad (5\text{-}12)$$

where k_p = reaction rate constant, mol/ l min atm^2

y_{A_o} = inlet mole fraction of A

P = pressure, atm

V = volume, l

K_p = equilibrium constant, atm^{-1}

N_{A_o} = inlet flow rate of A, mole/min

a, b, c = constants

For the following values,

F_{A_o} = 0.60 kmol/min

F_{B_o} = 1.20 kmol/min

F_{R_o} = 0.0

F_{I_o} = 1.0 kmol/min

k_c = 133.2 l/mol min

K_c = 1.95 l/mol

P = 10 atm

T = 300 °C

the program calculates a volume of 181.7 liters for a conversion of 10%. After converting rate constants to a pressure basis, calculating y_{A_o}, and the constants a, b, and c, Equation (5-12) was graphically integrated and the reactor volume calculated as 179.9 liters. The result calculated by the program, 181.7 liters, agrees with the evaluation of Equation (5-12).

Test Case IV. For a simple reaction scheme, Equation (3-13) can be integrated analytically to determine the size of a recycle reactor needed to accomplish a given conversion. The following hypothetical data were used in the simulator to determine the size of a recycle reactor needed to accomplish a 90% single-pass conversion of reactant A.

$A \longrightarrow R$ $\qquad -r_A = k_c C_A$

$T = 100\ ^{o}C$ $\qquad k_c = 125\ min^{-1}$

$P = 1$ atm

The recycle stream was specified to be 30% of the reactor effluent stream. Thus R, the recycle ratio, is calculated as 0.30/0.70 = 0.4286.

The results of the simulation, presented in Appendix A, show that the reactor volume is 8058 l. The results of the analytical integration, presented in Appendix B, show that a volume of 8065 l is needed. This difference, -.09%, is attributable largely to the stream convergence tolerance used and to a lesser extent the errors introduced during the simulation's numerical integration.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

The purpose of this study was to develop a Chemical Process Simulator capable of modeling an ideal, plug flow reactor and various general reaction schemes. Use of this program is directed primarily towards undergraduate Chemical Reaction Engineering students and should serve to reinforce principles presented in the classroom.

The following conclusions can be drawn based upon the evaluation of the program and the previous discussion:

1. The program can accurately simulate any non-reactive mixing and splitting process for up to nine streams, five units, and nine components.

2. The plug flow reactor module is accurately represented based upon tests comparing reactor volumes calculated by the program to results obtained in the literature.

3. Comparisons between different reactor operations can be made quickly to determine the best or most desirable operating scheme.

The following recommendations are suggested for further work:

1. The addition of a batch reactor and a continuous stirred tank reactor as available unit modules is an obvious extension of this work. These additions would allow comparison of reactor size requirements between individual reactor types.

2. Expansion of the data bank to include more chemical compounds

would allow a greater variety of reactions between real components to be studied, rather than reacting hypothetical components such as "A" or "R".

3. Adding a function generator subroutine and a subroutine that accepts rate data, would allow reactions to be studied that are not represented by simple power-law kinetics.

4. Graphics output would make the program more interesting to use. The program would first have to be transferred to a system with graphics capabilities.

A SELECTED BIBLIOGRAPHY

(1) Bitzer, D. "The Wide World of Computer-Based Education." Advances In Computers, 15 (1976), pp. 239-283.

(2) Uttal, W. R. "On Conversational Interaction." Programmed Learning and Computer-Based Instruction, (J. E. Coulson, ed.) New York: Wiley, 1962.

(3) Licklider, L. C. R. "Preliminary Experiment in Computer-Aided Instruction." Programmed Learning and Computer-Based Instruction, (J. E. Coulson, ed.) New York: Wiley, 1962.

(4) Kingery, R. A., R. D. Berg, and E. H. Schillinger. "A Computer in the Classroom." Men and Ideas in Engineering: Twelve Histories From Illinois, Urbana: University of Illinois Press, 1967.

(5) Computers and Education: An International Bibliography on Computers in Education. Amsterdam: International Federation on Information Processing, 1970.

(6) "CAI in Japan." ACM Sigcue Bulletin 1, No. 1, (1973), pp. 19-22.

(7) Lower, S., G. Gerhold, S. G. Smith, K. J. Johnson, and J. W. Moore. "Computer-Assisted Instruction in Chemistry." Journal of Chemical Education, 56 (4) (Apr., 1979), pp. 219-227.

(8) Mah, R. S. H., "Recent Development in Process Design," Symposium on Basic Questions of Design Theory, Columbia Univ., New York, 1974.

(9) Kehat, E., and M. Shacham, "Chemical Process Simulation Programs-1." Process Technology 18(1/2) (1973), p. 35.

(10) _____, "Chemical Process Simulation Programs-2, Partitioning and Tearing System Flowsheets." Process Technology, 18 (3) (1973), p. 115.

(11) Parker, Arthur L., and Richard R. Hughes. "Approximation Programming of Chemical Processes-1: Optimization of FLOWTRAN Models." Computers and Chemical Engineering, 5 (3) (1983), pp. 123-133.

(12) Motard, R. L., and D. M. Himmelblau. "Current Situation on the Use of Computers in the Education of Chemical Engineers." Computers and Chemical Engineering, 3 (1979), pp. 213-216.

(13) Calo, J. M., and R. P. Andres. "Use of Interactive Graphics-Based Software for Teaching Chemical Engineering Principles." Computers and Chemical Engineering, 5 (4) (1981), pp. 197-209.

(14) Eckart, C. A., T. T. Oberle, M. L. Gilbert, and R. S. Tomaskovic. "The Use of the Plato Computer System in Chemical Engineering." Computers and Chemical Engineering, 5 (4) (1981), pp. 221-213.

(15) Thompson, D. W. "CHEMOS-A PL/I Based List-Organized Process Modeling Program for Interactive Student Use." Computers and Chemical Engineering, 6 (1) (1982), pp. 27-38.

(16) Vasek, Vladimir, and Jiri Klemes. "Simulation Programming for Desk-Top Computer COMPUCORP 625 in Basic Language." Computers and Chemical Engineering, 7 (3) (1983), pp. 175-182.

(17) Shacham, M., and Michael B. Cutlip. "Educational Utilization of PLATO in Chemical Reaction Engineering." Computers and Chemical Engineering, 5 (4) (1981), pp. 215-224.

(18) Himmelblau, D. M. "Practical Experience Using Graphics in Teaching Process Analysis and Simulation." Computers and Chemical Engineering, 5 (4) (1981), pp. 187-195.

(19) Levenspiel, O. L. Chemical Reaction Engineering. 2nd ed., New York: Wiley, 1972.

(20) Holland, Charles D., and Rayford G. Anthony. Fundamentals of Chemical Reaction Engineering. New Jersey: Prentice-Hall, Inc., 1979.

(21) Chen, Ning Hsing. Process Reactor Design. Boston: Allyn and Bacon, Inc., 1983.

(22) Fogler, H. Scott, The Elements of Chemical Kinetics and Reactor Calculations (A Self-Paced Approach). New Jersey: Prentice-Hall, Inc., 1974.

(23) Smith, J. M. Chemical Engineering Kinetics. New York: McGraw-Hill Book Company, Inc., 1956.

(24) Crowe, C. M., A. E. Hamielec, T. W. Hoffman, A. I. Johnson, and D. R. Woods. Chemical Plant Simulation - An Introduction to Computer-Aided Steady-State Process Analysis. New Jersey: Prentice-Hall, Inc., 1971.

(25) Henley, Ernest J., and Edward M. Rosen. Material and Energy Balance Computations. New York: John Wiley and Sons, 1969.

(26) Ely, J. F., National Bureau of Standards. National Engineering Laboratory. Boulder: (1980)

(27) Petzold, Linda R., and Alan C. Hindmarsh. ODEPACK Documentation. Mathematics and Statistics Division, L-316, Lawrence Livermore National Laboratory. California, 1982.

(28) Gupta, G. K., R. Sacks-Davis, and P. E. Tischer. "A Review of Recent Developments in Solving ODEs." Computing Surveys, 17 (1) (March, 1985), pp. 7-47.

(29) Gerald, Curtis F., and Patrick O. Wheatley. Applied Numerical Analysis. Massachusetts: Addison-Wesley Publishing Co., 1984.

(30) Robertson, H. H. "Solution of a Set of Rate Equations." Numerical Analysis. J. Walsh, ed. Washington, D. C.: Thompson Book Company, 1966.

(31) Seinfeld, John H., Leon Lapidus, and Myungkyu Hwang. "Review of Numerical Integration Techniques for Stiff Ordinary Differential Equations." Industrial and Engineering Chemistry Fundamentals, 9 (2) (1970), pp. 266-274.

(32) Chan, Yau Nam I., Irving Birnbaum, and Leon Lapidus. "Solution of Stiff Differential Equations and the Use of Imbedding Techniques." Industrial and Engineering Chemistry Fundamentals, 17 (3) (1978), pp. 133-148.

(33) Guertin, Earl W. "Expontential Collocation of Stiff Reactor Models." Computers and Chemical Engineering, 1 (1977), pp. 197-203.

(34) White, Charles W., and Warren D. Seider. "Integration of Combustion Reaction Systems." Computers and Chemical Engineering, 8 (6) (1984), pp. 345-354.

(35) Hill, Charles G. An Introduction to Chemical Engineering Kinetics and Reactor Design. New York: John Wiley & Sons, 1977.

APPENDIXES

APPENDIX A

LISTING OF SIMULATION FOR PLUG FLOW REACTOR WITH RECYCLE

```
ENTER USER NAME.
BILL VEDDER

ENTER TITLE

RECYCLE EXAMPLE

PROCESS TOPOLOGY WILL NOW BE SET UP.
ENTER THE TOTAL NUMBER OF UNITS THAT WILL BE USED IN
     THE SIMULATION.
3


UNIT ID'S WILL NOW BE ENTERED.
ENTER UNIT ID'S IN THE DESIRED CALCULATIONAL ORDER.


UNITS AVAILABLE ARE:          UNIT          UNIT ID
                          ----------------------------
                            STREAM MIXER        MX
                          STREAM SPLITTER       SP
                        PLUG FLOW REACTOR       PF
(NOT AVAILABLE)       STIRRED TANK REACTOR      CS
(NOT AVAILABLE)             BATCH REACTOR       BR

ENTER UNIT ID FOR UNIT  1
MX

ENTER UNIT ID FOR UNIT  2
PF

ENTER UNIT ID FOR UNIT  3
SP

STREAM IDENTIFICATION NUMBERS WILL NOW BE ENTERED.
THE MAXIMUM ALLOWABLE STREAM ID NUMBER IS 9.
THERE IS NO CHECK ON THIS VARIABLE.


ENTER STREAM NUMBERS FOR ALL STREAMS ENTERING UNIT 1- MIXER
1,3

ENTER STREAM NUMBERS FOR ALL STREAMS LEAVING UNIT 1- MIXER
2

ENTER STREAM NUMBERS FOR ALL STREAMS ENTERING UNIT 2- PLUG FLOW REA
2

ENTER STREAM NUMBERS FOR ALL STREAMS LEAVING UNIT 2- PLUG FLOW REA
4
```

```
ENTER STREAM NUMBERS FOR ALL STREAMS ENTERING UNIT 3- SPLITTER
4

ENTER STREAM NUMBERS FOR ALL STREAMS LEAVING UNIT 3- SPLITTER
5,3

DO YOU WANT TO CHECK THE PROCESS TOPOLOGY? (Y/N)

```

UNIT #	UNIT ID	ASSOCIATED STREAMS				
1	MX	1	3	-2	0	0
2	PF	2	-4	0	0	0
3	SP	4	-5	-3	0	0

```
IS EVERYTHING OK? (Y/N)


ENTER FLUID PHASE.   1) GAS    2) LIQUID (NOT AVAILABLE)

1

ENTER SYSTEM PRESSURE (ATM)
1


PROCESS FEED STREAM DATA WILL NOW BE ENTERED.
THERE IS A MAXIMUM OF NINE COMPONENTS PER STREAM
AND NINE DIFFERENT COMPONENTS IN THE PROCESS.


ENTER TEMPERATURE (C) FOR PROCESS FEED STREAM   1

100

ENTER ID NUMBERS AND FLOWRATES (KG-MOLES/MIN) FOR COMPONENTS IN STREAM 1.
ENTER 0,0 TO TERMINATE INPUT FOR EACH STREAM.
1,10

0,0

KINETIC DATA WILL NOW BE ENTERED.
UP TO 14 CHARACTERS PER REACTION CAN BE ACCEPTED.
```

```
ENTER REACTION TYPE FOR REACTION # 1:      1) ELEMENTARY   2) NON-ELEMENTARY



ENTER REACTION #  1
HIT <RETURN> TO TERMINATE DATA ENTRY.
1A>1R

ENTER COMPONENT ID # FOR COMPONENT  A
1

ENTER COMPONENT ID # FOR COMPONENT  R
2

ENTER REACTION TYPE FOR REACTION # 2:      1) ELEMENTARY   2) NON-ELEMENTARY



ENTER REACTION #  2
HIT <RETURN> TO TERMINATE DATA ENTRY.


ENTER RECYCLE TOLERANCE (.1-.000001). SUGGESTED VALUE
IS .00001.
.00001


SELECT UNIT # 1-MIXER    OPERATING CONDITION:
    1) ISOTHERMAL
    2) NONISOTHERMAL

1


SELECT UNIT # 2-PLUG FLO OPERATING CONDITION:
    1) ISOTHERMAL
    2) NONISOTHERMAL

1

ENTER CONVERSION FOR UNIT #  2-  PFR
.90

ENTER RATE CONSTANT (L-MOLE-MIN) FOR FORWARD REACTION 1
125

PLEASE SPECIFY FRACTION OF FEED STREAM TO PRODUCT STREAM #  5
.70
```

```
WOULD YOU LIKE TO SEE A PROCESS MAP BEFORE SIMULATION? (Y,N)

   UNIT #     UNIT ID        ASSOCIATED  STREAMS
-------------------------------------------------------
     1          MX          1     3    -2     0     0
     2          PF          2    -4     0     0     3
     3          SP          4    -5    -3     0     0

FOR UNIT # 1 -MX  OPERATION IS ISOTHERMAL

FOR UNIT # 2 -PF  OPERATION IS ISOTHERMAL

FOR UNIT # 3 -SP  OPERATION IS ISOTHERMAL

 SP UNIT #    PROD STRM#    FRACN OF FEED STRM
      3           5              0.70
      3           3              0.30
FOR UNIT # 2 PF CONVERSION IS 0.90


IS THE SIMULATION READY TO RUN? (Y/N)


>>>>>>>>>>  SIMULATION EXECUTING  <<<<<<<<<<

 . 3.>>>>>>>>   UNIT # 2 HAS CONVERGED


>>>>>>>>>>>>   UNIT # 3 HAS CONVERGED


>>>>>>>>>>>>   UNIT # 1 HAS CONVERGED


PRINT RESULTS? (Y/N)
```

NAME:
TITLE: RECYCLE EXAMPLE

********** STREAM NUMBER 1 **********
TEMPERATURE (C) : 100.0000

CMP	MOL FLOW KG-MOLS/MIN	MASS FLOW KG/MIN	X	H MCAL	W
H2	10.0000	20.2000	1.0000	25.4603	1.0000

********** STREAM NUMBER 2 **********
TEMPERATURE (C) : 100.0000

CMP	MOL FLOW KG-MOLS/MIN	MASS FLOW KG/MIN	X	H MCAL	W
H2	10.3097	20.8247	0.7216	26.2477	0.2461
CH4	3.9764	63.7938	0.2784	12.2603	0.7539

********** STREAM NUMBER 3 **********
TEMPERATURE (C) : 100.0000

CMP	MOL FLOW KG-MOLS/MIN	MASS FLOW KG/MIN	X	H MCAL	W
H2	0.3093	0.6247	0.0722	0.7874	0.0097
CH4	3.9764	63.7938	0.9278	12.2603	0.9903

********** STREAM NUMBER 4 **********
TEMPERATURE (C) : 100.0000

CMP	MOL FLOW KG-MOLS/MIN	MASS FLOW KG/MIN	X	H MCAL	W
H2	1.0309	2.0825	0.0722	2.6248	0.0097
CH4	13.2548	212.6464	0.9278	40.8676	0.9903

```
**********************  STREAM NUMBER 5 **********************
TEMPERATURE (C) :    100.0000

 CMP        MOL FLOW        MASS FLOW          X            H            W
          KG-MOLS/MIN        KG/MIN                        MCAL

H2           0.7216           1.4577        0.0722        1.2373       0.0097
CH4          9.2797         148.8622        0.9278       23.6373       0.9903


UNIT # 2   UNIT: PF REACTOR VOLUME (L) :       8059.031
                          CONVERSION :          0.900

REACTIONS:   1)1A>1R

DO YOU WISH TO VIEW YIELD DATA?  (Y/N)
N


DO YOU WISH TO VIEW DATA GENERATED DURING INTEGRATION? (Y/N)
N



WOULD YOU LIKE TO MAKE ANOTHER RUN? (Y/N)
N

STATEMENTS EXECUTED=  233709
CORE USAGE       OBJECT CODE=  212864 BYTES,ARRAY AREA=   49268 BYTES,TOTAL AREA AVAILABLE=  [illegible]
DIAGNOSTICS        NUMBER OF ERRORS=       0, NUMBER OF WARNINGS=       0, NUMBER OF EXTENSIONS=
COMPILE TIME=     0.91 SEC,EXECUTION TIME=     1.30 SEC,      9.19.45    SATURDAY      7 DEC 85
        C$STOP
```

APPENDIX B

ANALYTICAL DEVELOPMENT AND SOLUTION OF PLUG FLOW REACTOR WITH RECYCLE TEST CASE IV

For the elementary reaction

$$A \xrightarrow{k_c} R \tag{B-1}$$

the rate of disappearance of A is

$$-r_a = k_c\, C_A \tag{B-2}$$

The concentration of A is expressed in terms of the simulation input data and conversion as

$$C_A = \frac{F_a}{v} = \frac{F_{A_o}\,(1 - X_A)}{v} \tag{B-3}$$

Equation (3-13) can now be written as

$$V = F_{a_o}(R+1) \int_{(\frac{R}{R+1})X_{A_f}}^{X_{A_f}} \frac{dX_A}{k_c \dfrac{F_{A_o}\,(1-X_A)}{v}} \tag{B-4}$$

Since the simulator requires input of the single-pass conversion, X_{A_f} must be calculated from the results presented in APPENDIX A for the flow rate of component A in streams 1 (process feed) and 5 (process effluent).

$$X_{A_f} = 1. - \frac{F_A}{F_{A_o}} = 1. - \frac{.7216}{10.0} \tag{B-5}$$

$$X_{A_f} = 0.928 \tag{B-6}$$

Equation (B-4) can now be evaluated.

$$V = \frac{v\ (R+1)}{k_c} \int_{0.928}^{0.2784} \frac{dX_A}{1-X_A} \tag{B-7}$$

$$V = \frac{306{,}207\ \text{l/min}\ (1.4286)}{125\ \text{min}^{-1}}\ \ln\left\{\frac{1-0.2784}{1-0.928}\right\} \tag{B-8}$$

$$V = 8066\ \text{liters} \tag{B-9}$$

APPENDIX C

KINETX USERS MANUAL

Overview

KINETX is an interactive, chemical process simulator designed to be used in conjunction with a senior-level reaction engineering course. Currently available unit modules are a stream splitter, stream mixer, and plug flow reactor. Up to five units and nine streams can be simulated. The maximum number of components per stream is nine and up to nine different components in a process can be simulated. A maximum of five reactions (reversible or irreversible) can be input. Isothermal or nonisothermal reactor operation can be specified. Thermodynamic data are available for the Gas Processors Association's 61 most used components. Currently, only gas phase processes can be simulated. KINETX is run on Oklahoma State University's IBM 3081K mainframe.

Description of Unit Modules

Mixer

The mixer module simulates an adiabatic mixing process. It can accept up to four feed streams and will output one effluent stream. If the feed streams are entering the mixer at different temperatures, nonisothermal operation is specified.

Splitter

The splitter module simulates an adiabatic, isothermal stream splitting process. It will accept only one feed stream and output a

maximum of four effluent streams. KINETX requests the user to specify the fraction of the total feed that is to flow to the outlet streams. The user inputs a decimal fraction between 0.0 and 1.0. KINETX makes this request N-1 times (N is the number of outlet streams from the splitter) then computes the fraction for the N-th stream automatically.

Plug Flow Reactor (PFR)

The PFR module simulates an ideal, gas phase plug flow reactor. Only one inlet and one outlet stream can be specified. The PFR can be operated either isothermally or nonisothermally. KINETX requests the user to specify the conversion for the PFR unit (ie. X_A). An important point to note is that this is the single-pass conversion of the key component in the reactor and not the overall conversion. This distinction is only necessary for recycle problems.

Simulation Procedure Description

Input Data Required

Prior to the actual computer work, the user should have a flowsheet of the process to be simulated. Each stream should be numbered consecutively from one to N (where N is the total number of streams in the process and N is $\geqq$ nine). Process feed stream data must be known. These data are temperatures (^{o}C), component identification numbers (from Table III), and component flow rates (kg-mol/min). The system pressure (atm) must also be known. Kinetic data required are the reaction(s) to be run, component identification numbers (from Table III), and the type of reaction (elementary or nonelementary). For isothermal reactor operation, the program requests rate constants

TABLE III

LIST OF COMPONENTS

Component Id. No.	Name	Program Symbol
1	Hydrogen	H_2
2	Methane	CH_4
3	Ethane	C_2H_6
4	Propane	C_3H_8
5	iso-Butane	$I\text{-}C_4H_{10}$
6	n-Butane	$N\text{-}C_4H_{10}$
7	iso-Pentane	$I\text{-}C_5H_{12}$
8	n-Pentane	$N\text{-}C_5H_{12}$
9	neo-Pentane	$NEO\text{-}C_5$
10	n-Hexane	$N\text{-}C_6H_{14}$
11	n-Heptane	$N\text{-}C_7H_{16}$
12	n-Octane	$N\text{-}C_8H_{18}$
13	n-Nonane	$N\text{-}C_9H_{20}$
14	n-Decane	$N\text{-}C_{10}H_{22}$
15	n-Undecane	$N\text{-}C_{11}H_{24}$
16	n-Dodecane	$N\text{-}C_{12}H_{26}$
17	n-Tridecane	$N\text{-}C_{13}H_{28}$
18	n-Tetradecane	$N\text{-}C_{14}H_{30}$
19	n-Pentadecane	$N\text{-}C_{15}H_{32}$

TABLE III (Continued)

Component Id. No.	Name	Program Symbol
20	n-Hexadecane	$N\text{-}C_{16}H_{34}$
21	n-Heptadecane	$N\text{-}C_{17}H_{36}$
22	Ethylene	$C_2H_4=$
23	Propylene	$C_3H_6=$
24	1-Butene	$1\text{-}C_4H_8$
25	cis-2-Butene	$C\text{-}2\text{-}C_4H_8$
26	trans-2-Butene	$T\text{-}2\text{-}C_4H_8$
27	iso-Butene	$1\text{-}C_4H_8$
28	1,3 Butadiene	$1,3\text{-}C_4==$
29	1-Pentene	$1\text{-}C_5H_{10}$
30	cis-2-Pentene	$C\text{-}2\text{-}C_5=$
31	trans-2-Pentene	$T\text{-}2\text{-}C_5=$
32	2-Methyl-1-Butene	$2MT\text{-}1C_4=$
33	3-Methyl-1-Butene	$3MT\text{-}1C_4=$
34	2-Methyl-2-Butene	$2MT\text{-}2C_4=$
35	1-Hexene	$C_6H_{12}=$
36	Cyclopentane	$CYC\text{-}C_5$
37	Methylcyclopentane	$MTCYC\text{-}C_6$
38	Cyclohexane	$CYC\text{-}C_6$
39	Methylcyclohexane	$MTCYC\text{-}C_6$
40	Benzene	BZ
41	Toluene	TOL
42	o-Xylene	O-X

TABLE III (Continued)

Component Id. No.	Name	Program Symbol
43	m-Xylene	M-X
44	p-Xylene	P-X
45	Ethylbenzene	EB
46	Nitrogen	N_2
47	Oyxgen	O_2
48	Carbon Monoxide	CO
49	Carbon Dioxide	CO_2
50	Hydrogen Sulfide	H_2S
51	Sulfur Dioxide	SO_2
52	2-methyl-Pentane	2-MT-C_5
53	3-methyl-Pentane	3-MT-C_5
54	2,2 dimethyl-Butane	2,2 $DMTC_4$
55	2,3 dimethyl-Butane	2,3 $DMTC_4$
56	1-Heptene	1-C_7H_{14}=
57	Propadiene	C_3H_4==
58	1,2, Butadiene	1,2-C_4==
60	Ethylcyclohexane	ETCYC-C_6
61	Water	H_2O

(l-mol-min) and equilibrium constants (for reversible reactions). For nonisothermal reactor operation the program requests Arrhenius constants (l-mol-min), activation energies (cal/mol), heats of reaction (cal/mol), and equilibrium constants (for reversible reactions). Since heats of reaction and equilibrium constants are assumed independent of temperature, these data should be valid for an estimated average temperature or, at least, the inlet temperature to the reactor. If a nonelementary reaction has been specified the exponents on the concentration terms in the rate expression must be known. If a splitter module is part of the simulation, the fraction of the feed stream that is to flow to the product streams must be known.

Explanations of Data Requests

This section will explain the requests for data that are made by KINETX. Most of the input data are checked for validity. When invalid data are entered or an error in the process topology has been detected (such as specifying more than one feed stream to a splitter) an appropriate message is issued and a request for new data is made. Throughout this section, the requests made by the program will be shown as dot matrix print while explanations are shown in regular type. To begin a simulation, the following command is issued in the READY mode:

```
CALL 'U14319A.PROCSIM.LOAD(KINETX)'
```

The simulator is now running. Input data requests and explanations are as follows:

```
ENTER USER NAME.
```

A maximum of 20 characters can be entered for the user's name.

```
ENTER TITLE
```

A maximum of 60 characters can be entered for the title of the simulation.

```
PROCESS TOPOLOGY WILL NOW BE SET UP.
ENTER THE TOTAL NUMBER OF UNITS THAT WILL BE USED IN
     THE SIMULATION.
```

The total number of units is the number of mixers, splitters, and plug flow reactors in the simulation. Respond with an integer number no greater than five.

```
UNIT ID'S WILL NOW BE ENTERED.
ENTER UNIT ID'S IN THE DESIRED CALCULATIONAL ORDER.

UNITS AVAILABLE ARE:
```

UNIT	UNIT ID
STREAM MIXER	MX
STREAM SPLITTER	SP
PLUG FLOW REACTOR	PF
STIRRED TANK REACTOR	CS
BATCH REACTOR	BR

```
ENTER UNIT ID FOR UNIT  1
```

Enter the two character unit ID mnemonic corresponding to the desired unit. This request is repeated N times where N is the total number of units in the process.

```
STREAM IDENTIFICATION NUMBERS WILL NOW BE ENTERED.
THE MAXIMUM ALLOWABLE STREAM ID NUMBER IS 9.
THERE IS NO CHECK ON THIS VARIABLE.

ENTER STREAM NUMBERS FOR ALL STREAMS ENTERING UNIT 1- MIXER
```

Respond with the stream numbers identifying all of the feed streams

to unit 1. These are the numbers from the flowsheet. If more than one stream is entering a unit, separate each stream number with a comma. When all the stream numbers have been listed, press the RETURN key.

ENTER STREAM NUMBERS FOR ALL STREAMS LEAVING UNIT 1- MIXER

Respond with the stream numbers identifying the effluent streams from unit 1, in the same manner as above.

Both of the above requests for stream numbers are repeated for each unit in the process.

DO YOU WANT TO CHECK THE PROCESS TOPOLOGY? (Y/N)

Respond with either a 'Y' or simply hit RETURN to check the topology. If this is done, the matrix representation of the process will appear. For the process shown in Figure 4, the following matrix would appear:

UNIT #	UNIT ID	ASSOCIATED STREAMS				
1	MX	1	6	-2	0	0
2	PF	2	-3	0	0	0
3	MX	3	4	-5	0	0
4	SP	5	-4	-7	0	0
5	PF	7	-8	0	0	0

Feed streams to each unit are represented as positive numbers while the effluent streams are represented as negative numbers. Check this matrix against your flowsheet. If everything is correct, enter a 'Y' or simply hit RETURN. If you enter 'N' the topology must be reentered. A comment regarding error checks is in order here. Since the program checks for errors in the topology (such as having more than one feed stream to a splitter) after the above data are shown, an error message

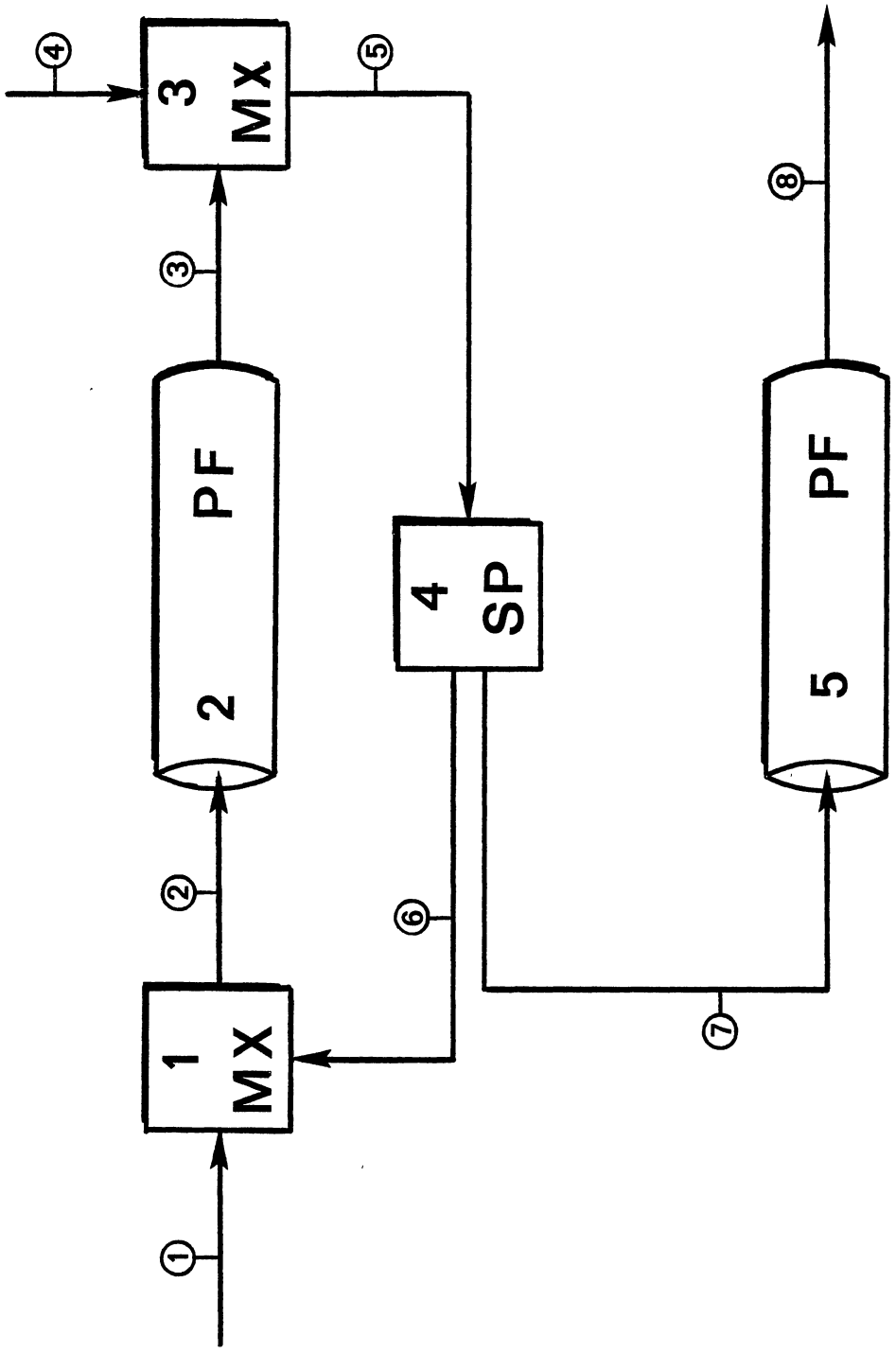

Figure 4. Example Flowsheet

may still be issued even though you have responded that everything is OK. The error message will state the nature of the problem. Your flowsheet needs to be altered accordingly.

```
ENTER FLUID PHASE.   1) GAS      2) LIQUID
```

Respond by entering a '1'.

```
ENTER SYSTEM PRESSURE (ATM)
```

Respond by entering the pressure of the system. Since pressure is assumed constant, this is the pressure throughout the entire system.

```
PROCESS FEED STREAM DATA WILL NOW BE ENTERED.
THERE IS A MAXIMUM OF NINE COMPONENTS PER STREAM
AND NINE DIFFERENT COMPONENTS IN THE PROCESS.

ENTER TEMPERATURE (C) FOR PROCESS FEED STREAM  1
```

KINETX has identified stream number 1 as a feed to the process, therefore stream 1 will have to be defined. Start by entering the temperature for this stream.

```
ENTER ID NUMBERS AND FLOWRATES (KG-MOLES/MIN) FOR COMPONENTS IN STREAM 1.
ENTER 0,0 TO TERMINATE INPUT FOR EACH STREAM.
```

Enter the component identification number (from Table III) and flow rate separated by a comma, then press the RETURN key. Do this for each component, then enter 0,0 to proceed.

Respond by entering a '1' or simply hit RETURN to specify an elementary reaction or enter a '2' to specify a nonelementary reaction.

ENTER REACTION # 1

There are conventions that must be followed in order to successfully respond to the above request for a chemical reaction equation. These are:

1. The only acceptable characters are alphanumeric (A-Z and 1-9), the '+' sign, the '>' sign (which signifies an irreversible reaction), and the '=' sign (which signifies a reversible reaction).

2. There can be no imbedded blanks in the reaction equation.

3. A strict sequence must be followed for inputting the characters in the reaction equation. The first character must be numeric, followed by an alphabetic character, which in turn is followed by a separator ('+', '>', '='). It is especially important to remember that this sequence must be followed even if the stoichiometric coefficient is a '1'. The '1' MUST be entered as part of the equation.

4. KINETX assumes that the key component for each reaction (the limiting reactant or the component for which the rate equation is written), is the first component in the equation. Therefore if the rate equation describes the rate of disappearance of 'A' ($-r_A$), the component 'A' should be the first alphabetic character in the reaction. For example, if the stoichiometric equation is:

$$2B + A \rightleftharpoons 3R$$

and the rate equation is written for 'A' such as:

$$-r_A = k_1 C_B^2 C_A - k_2 C_R^3$$

then the equation must be entered as:

1A+2B=3R

The requests for the type of reaction and the reaction itself are issued automatically. After each reaction is entered a request is made for the component I.D. number which identifies the component in the reaction. Respond by entering the appropriate I.D. number from TABLE III. To terminate the requests for kinetic data press the RETURN key for each request without entering any data.

```
ENTER RECYCLE TOLERANCE (.1-.000001). SUGGESTED VALUE
IS .00001.
```

Respond with an appropriate value between .1 and .000001.

```
SELECT UNIT # 1-MIXER    OPERATING CONDITION:
   1) ISOTHERMAL
   2) NONISOTHERMAL
```

In this example, a stream mixer is part of the simulation. If the feed streams to this unit are at different temperatures or one of the feeds is a recycle stream from a PFR that is operating nonisothermally, enter '1'. The above request is made once for each mixer and PFR in the simulation.

```
ENTER CONVERSION FOR UNIT #  2-  PFR
```

In this example, unit 2 is a PFR. Respond with the appropriate single-pass conversion of the key component.

```
ENTER RATE CONSTANT (L-MOLE-MIN) FOR FORWARD REACTION 1
```

For isothermal reactions, the above request for a rate constant is issued for each reaction. If the reaction is reversible (as is our example) a request for an equilibrium constant is made. For nonisothermal reactions KINETX requests the Arrhenius constant, heat of reaction, and activation energy for each forward reaction. Reversible reactions, again, require only the equilibrium constant. This effectively ignores the temperature dependency of the equilibrium constant. The dependency of the heat of reaction upon temperature is also ignored. To offset these assumptions somewhat, these data should be applicable at an estimated average temperature or at least the temperature of the feed to the reactor.

If a splitter module is part of the simulation KINETX requests the user to specify the fraction of the feed stream that is to flow to a specific outlet stream. This request is made N-1 times for each splitter in the process. N is the number of outlet streams from a splitter. In our example the request is:

PLEASE SPECIFY FRACTION OF FEED STREAM TO PRODUCT STREAM # 6

Respond with a number between 0.0 and 1.0. KINETX calculates the fraction to stream number 7.

After all the data is entered, KINETX will ask the user if he or she would like to see a process map before execution. At this point the user can check most of the input data and make changes. For our example, the user responded to the request by simply pressing the RETURN key.

WOULD YOU LIKE TO SEE A PROCESS MAP BEFORE SIMULATION ? Y

```
UNIT #     UNIT ID        ASSOCIATED STREAMS
-------------------------------------------------------------------
   1          MX          1    2    -3    0    0
   2          FF          3   -4     0    0    0
   3          MX          4    5    -6    0    0
   4          SP          6   -7    -8    0    0
   5          FF          8   -9     0    0    0

FOR UNIT # 1 -MX  OPERATION IS ISOTHERMAL

FOR UNIT # 2 -FF  OPERATION IS ISOTHERMAL

FOR UNIT # 3 -MX  OPERATION IS ISOTHERMAL

FOR UNIT # 4 -SP  OPERATION IS ISOTHERMAL

FOR UNIT # 5 -FF  OPERATION IS ISOTHERMAL

 SP UNIT #    PROD STRM#    FRACN OF FEED STRM
     4            6              0.90
     4            7              0.10
FOR UNIT # 2 FF CONVERSION IS 0.90
FOR UNIT # 5 FF CONVERSION IS 0.90
IS THE SIMULATION READY TO RUN? (Y/N)
```

If the data represent the process accurately, respond by entering 'Y' or RETURN. If changes need to be made, respond with 'N'. A menu will appear on the screen which will tell the user which data can be changed.

Three different data sets are generated during the simulation; stream data, yield data, and data generated during the integration of the rate equations. Stream data includes component molar and mass flow rates, component mole and mass fractions and component enthalpies. This data includes the reactor volume required to accomplish the desired conversion of the key component. Fractional yields for selected products can be output after responding to:

```
DO YOU WISH TO VIEW YIELD DATA? (Y/N)
```

The user enters the component identification numbers for those products whose yield data is desired by responding to:

ENTER COMPONENT ID NUMBER FOR CONVERSION:

Fractional yield based upon key component fed to the process and yield based upon the amount of the key component reacted will be output.

Data generated during the course of the reaction will show how component flows, temperature, and reactor volume vary as a function of conversion. This data can be seen by responding to:

DO YOU WISH TO VIEW DATA GENERATED DURING INTEGRATION? Y/N

After the data has been output, the user can choose to run another problem or exit the simulator.

WOULD YOU LIKE TO MAKE ANOTHER RUN? (Y/N)

Responding with a 'Y' or RETURN will start another simulation from the point of asking for the title of the problem.

APPENDIX D

PROGRAM LISTING

```
//U14319A JOB (14319,341-50-4556),' ',TIME=(0,10),                    00000100
// CLASS=A,MSGCLASS=X                                                  00000200
//  EXEC FORTVCL,REGION.FORT=1200K                                     00000300
//FORT.SYSIN DD *                                                     00000400
      COMMON/TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)   00001800
     *          ,NMRCTR                                              00001900
      COMMON /EDFLAG/NEDFLG(2)                                        00002000
      COMMON/INOUT/NW,NR                                              00002100
      COMMON /ID/TITLE,NAME                                           00002200
      CHARACTER TITLE*60,NY,NAME*20,IDUM*2                            00002300
      NW=6                                                            00002400
      NR=5                                                            00002500
      WRITE(NW,100)                                                   00002600
100   FORMAT(//' ENTER USER NAME. ')                                  00002700
      READ(NR,99) NAME                                                00002800
99    FORMAT(A20)                                                     00002900
10    CALL INISHL                                                     00003000
      WRITE(NW,101)                                                   00003100
101   FORMAT(' ENTER TITLE'/)                                         00003200
      READ(NR,102) TITLE                                              00003300
102   FORMAT(A60)                                                     00003400
      CALL TOPOLI                                                     00003500
      CALL STNSCM                                                     00003600
      CALL MXMNIO                                                     00003700
      CALL STRMI                                                      00003800
      IF(NMRCTR .GT. 0) CALL RXNI                                     00003900
      CALL STUNIT                                                     00004000
C                                                                     00004100
C   NOW MUST ALLOW THE USER TO CHECK THE WHOLE PROCESS UNIT BY UNIT.   00004200
C                                                                     00004300
      WRITE(NW,104)                                                   00004400
104   FORMAT(' WOULD YOU LIKE TO SEE A PROCESS MAP BEFORE SIMULATION?', 00004500
     *       ' (Y/N)')                                                00004600
      READ(NR,103) NY                                                 00004700
      IF(NY .EQ. 'Y' .OR. NY .EQ. ' ') CALL PRCMAP                    00004800
      WRITE(NW,110)                                                   00004900
      READ(NR,108) IDUM                                               00005000
      IF(IDUM .EQ. 'Y' .OR. IDUM .EQ. ' ') GOTO 15                    00005100
      CALL EDIT                                                       00005200
      IF(NEDFLG(1) .EQ. 1) GOTO 15                                    00005300
      GOTO 10                                                         00005400
15    WRITE(NW,105)                                                   00005500
105   FORMAT(' >>>>>>>>>>   SIMULATION EXECUTING   >>>>>>>>>>')       00005600
      CALL EXEC                                                       00005700
      CALL OUTPUT                                                     00005800
110   FORMAT(//' IS THE SIMULATION READY TO RUN? (Y/N)')              00005900
108   FORMAT(A1)                                                      00006000
      WRITE(NW,111)                                                   00006100
111   FORMAT(///' WOULD YOU LIKE TO MAKE ANOTHER RUN? (Y/N)')         00006200
      READ(NR,103) NY                                                 00006300
103   FORMAT(A1)                                                      00006400
      IF(NY .EQ. ' ' .OR. NY .EQ. 'Y') GOTO 10                        00006500
      STOP                                                            00006600
      END                                                             00006700
      SUBROUTINE INTDTA                                               00006710
C                                                                     00006720
C*********************************************************************00006730
C                                                                     00006740
C   THE PURPOSE IF THIS ROUTINE IS TO PRINT DATA THAT WAS GENERATED   00006750
C   DURING THE COURSE OF THE INTEGRATION. THIS DATA CONSISTS OF       00006760
C   COMPONENT FLOW RATES,TEMPERATURE,RATE OF "A", AND REACTOR VOLUME  00006770
C   AS A FUNCTION OF CONVERSION.                                      00006780
C                                                                     00006790
C*********************************************************************00006791
C                                                                     00006792
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FA0,P,VFLO              00006793
     *         ,IPHASE,MAXCMP,INERT(5)                                00006794
      COMMON /LIBDAT/NLIB,NDUM,CPROPS(22,61)                          00006795
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)  00006796
     *          ,NMRCTR                                               00006797
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                 00006798
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS 00006799
```

```
     *        ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)        00006800
     *        ,RXN(5,14)                                                 00006801
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00006802
     *             , ITRS,ITRT,MAXITR,CTOL                              00006803
      COMMON /RXSTRM/TOTFLO,RATE,RA                                     00006804
      COMMON /INOUT/NW,NR                                               00006805
      INTEGER FLGERR,UNTFLG                                             00006806
      CHARACTER RXN                                                     00006807
      REAL*8 CPROPS                                                     00006808
      DOUBLE PRECISION TOTFLO,RATE(5),FLONRT                            00006809
      WRITE(NW,100)                                                     00006812
100   FORMAT('***************   COMPONENT FLOWS (KG-MOLS'               00006813
     *,'/MIN)   **************     CONVERSION'/)                        00006814
C                                                                       00006815
C   SET N = NUMBER OF REACTIVE COMPONENTS IN THE SIMULATION             00006816
C                                                                       00006817
      N=NRXNTS                                                          00006818
      NN=0                                                              00006819
      WRITE(NW,101) (CPROPS(8,K),CPROPS(9,K),K=1,N)                     00006822
101   FORMAT(2X,18(A4),' CONVERSION'/)                                  00006823
      DO 11 I=1,5                                                       00006824
         DO 12 J=1,100                                                  00006825
            IF(RCT(I,J,1) .LE. 0.) GOTO 11                              00006826
12          WRITE(NW,102) (RCT(I,J,K),K=1,N),RCT(I,J,11)                00006827
11    CONTINUE                                                          00006828
102   FORMAT(1X,5(F10.3))                                               00006829
      WRITE(NW,103)                                                     00006835
103   FORMAT('  REACTOR VOLUME (L)     1/(-RA) (L-MOL-MIN)       XA'/)  00006836
      DO 14 I=1,5                                                       00006837
         DO 15 J=2,100                                                  00006838
            IF(RCT(I,J,1) .LE. 0.) GOTO 16                              00006839
15          WRITE(NW,104) RCT(I,J,12),RCT(I,J,14),RCT(I,J,11)           00006840
14    CONTINUE                                                          00006841
104   FORMAT(1X,F12.4,12X,F12.4,20X,F6.3)                               00006842
16    IF(RCT(1,3,13) .LE. 0.) GOTO 19                                   00006843
      WRITE(NW,105)                                                     00006844
105   FORMAT('  TEMPERATURE (K)       XA')                              00006845
      DO 17 I=1,5                                                       00006846
         DO 18 J=2,100                                                  00006847
            IF(RCT(I,J,1) .LE. 0.) GOTO 19                              00006848
18          WRITE(NW,106) RCT(I,J,13),RCT(I,J,11)                       00006849
17    CONTINUE                                                          00006850
106   FORMAT(2X,F10.3,8X,F6.3)                                          00006851
19    RETURN                                                            00006852
      END                                                               00006853
      SUBROUTINE STUNIT                                                 00006860
C                                                                       00006900
C***********************************************************************00007000
C                                                                       00007100
C   THIS ROUTINE WILL REQUEST ALL THE PARAMETERS AND OPERATING          00007200
C   CONDITIONS ASSOCIATED WITH EACH UNIT.                               00007300
C                                                                       00007400
C***********************************************************************00007500
C                                                                       00007600
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00007700
     *        ,IPHASE,MAXCMP,INERT(5)                                   00007800
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00007900
     *           ,NMRCTR                                                00008000
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                   00008100
     *        ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS   00008200
     *        ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00008300
     *        ,RXN(5,14)                                                00008400
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00008500
     *             , ITRS,ITRT,MAXITR,CTOL                              00008600
      COMMON /EDFLAG/NEDFLG(2)                                          00008700
      INTEGER FLGERR,UNTFLG                                             00008800
      CHARACTER RXN                                                     00008900
      DOUBLE PRECISION TOTFLO,FLONRT                                    00009000
      DIMENSION NOUT(5),NIN(5)                                          00009100
      DO 10 I=1,MAXUID                                                  00009200
         CALL NOUTO(NOUT)                                               00009300
         CALL NOUTO(NIN)                                                00009400
```

```
         CALL FNDOUT(I,NOUT,NUMOUT)                                          00009500
         CALL FNDIN(I,NIN,NUMIN)                                             00009600
         IF(NPM(I,2) .EQ. 1) THEN                                            00009700
            CALL STOPCD(I)                                                   00009800
            IF(NOPCND(I) .EQ. 2) CALL STTEMP(I)                              00009900
            CALL FNDREC(NIN,NUMIN)                                           00010000
         END IF                                                              00010100
         IF(NPM(I,2) .EQ. 2) THEN                                            00010200
            NOPCND(I)=1                                                      00010300
            CALL SPPRMS(I,NOUT,NUMOUT)                                       00010400
            IF(NOPCND(I) .EQ. 2) CALL STTEMP(I)                              00010500
         END IF                                                              00010600
         IF(NPM(I,2) .EQ. 3) THEN                                            00010700
            CALL STOPCD(I)                                                   00010800
            CALL STX(I)                                                      00011000
            IF( NOPCND(I) .EQ. 2) THEN                                       00011100
               CALL STE                                                      00011200
               CALL STA                                                      00011300
               CALL STDELH                                                   00011400
            ELSE                                                             00011500
               CALL STRK                                                     00011600
            END IF                                                           00011700
         END IF                                                              00011800
10    CONTINUE                                                               00011900
      RETURN                                                                 00012000
      END                                                                    00012100
      SUBROUTINE YIELD                                                       00012110
C                                                                            00012120
C*************************************************************************** 00012130
C                                                                            00012140
C   THE PURPOSE OF THIS ROUTINE IS TO COMPUTE YIELD DATA FOR USER            00012150
C   SPECIFIED PRODUCTS. TWO YIELDS ARE USED. THESE ARE:                      00012160
C                                                                            00012170
C       FRACTIONAL YIELD = FY = THIS IS THE MOLES OF PRODUCT PRODUCED        00012180
C                        PER MOLE KEY COMPONENT REACTED                      00012190
C                                                                            00012191
C       FRACTIONAL YIELD (BASED ON FEED) = FYBOF = THIS IS THE MOLES OF      00012192
C                                         PRODUCT PRODUCED PER MOLE OF       00012193
C                                         KEY COMPONENT FED TO SYSTEM.       00012194
C                                                                            00012195
C*************************************************************************** 00012196
C                                                                            00012197
      COMMON /INOUT/NW,NR                                                    00012198
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                     00012199
     *         ,IPHASE,MAXCMP,INERT(5)                                       00012200
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)         00012201
     *           ,NMRCTR                                                     00012202
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                           00012203
     *            CMP(100)                                                   00012204
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                        00012205
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS       00012206
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)           00012207
     *         ,RXN(5,14)                                                    00012208
      COMMON /LIBDAT/NLIB,NDUM,CPROPS(22,61)                                 00012209
      REAL*8 CPROPS                                                          00012210
      CHARACTER RXN                                                          00012211
      CHARACTER CHRCHK,SEP,MODULE*8,IDUM,CHRUID*2                            00012212
      DOUBLE PRECISION FLONRT                                                00012213
      INTEGER OS,YLDCMP(5)                                                   00012214
      DIMENSION IS(5),OS(5),FY(5),FYBOF(5),YCMPFL(5)                         00012215
      DO 10 I=1,5                                                            00012216
         YLDCMP(I)=0                                                         00012217
         YCMPFL(I)=0.                                                        00012218
         FY(I)=0.                                                            00012219
         FYBOF(I)=0.                                                         00012220
         IS(I)=0.                                                            00012221
10       OS(I)=0.                                                            00012222
      WRITE(NW,100)                                                          00012223
100   FORMAT(/' ENTER THE NUMBER OF COMPONENTS FOR WHICH YIELD DATA IS'      00012224
     *,' DESIRED.')                                                          00012225
      READ(NR,*) NUMCMP                                                      00012226
      CALL IERRLM(1,5,NUMCMP)                                                00012227
```

```
      WRITE(NW,102)                                                     00012228
102   FORMAT(' COMPONENT ID NUMBERS WILL NOW BE ENTERED FOR THOSE ',    00012229
     * 'COMPONENTS FOR WHICH'/' YIELD DATA IS DESIRED.'/)               00012230
      DO 11 I=1,NUMCMP                                                  00012231
         WRITE(NW,101) I                                                00012232
101      FORMAT(' ENTER COMPONENT ID NUMBER FOR COMPONENT',I2)          00012233
         READ(NR,*) YLDCMP(I)                                           00012234
         CALL IERRLM(1,61,YLDCMP(I))                                    00012235
11    CONTINUE                                                          00012236
C                                                                       00012237
C   FIND ALL PROCESS FEED AND EFFLUENT STREAMS.                         00012238
C                                                                       00012239
      KI=0                                                              00012240
      KO=0                                                              00012241
      DO 12 I=1,9                                                       00012242
         IF(NSCM(I,1) .EQ. 0) GOTO 15                                   00012243
         IF(NSCM(I,2) .EQ. 0) GOTO 13                                   00012244
         IF(NSCM(I,3) .EQ. 0) GOTO 14                                   00012245
         GOTO 12                                                        00012246
13       KI=KI+1                                                        00012247
         IS(KI)=NSCM(I,1)                                               00012248
         GOTO 12                                                        00012249
14       KO=KO+1                                                        00012250
         OS(KO)=NSCM(I,1)                                               00012251
12    CONTINUE                                                          00012252
C                                                                       00012253
C   FIND TOTAL AMOUNT OF "A" THAT IS FED TO THE PROCESS                 00012255
C                                                                       00012256
15    FLOAI=0.                                                          00012257
      DO 16 I=1,KI                                                      00012258
         DO 17 J=1,9                                                    00012259
            IF(INT(STRMID(IS(I),J,1)) .EQ. KEYPOS(1)) THEN              00012261
               FLOAI=FLOAI+STRMID(IS(I),J,2)                            00012262
               GOTO 16                                                  00012264
            END IF                                                      00012265
17       CONTINUE                                                       00012266
16    CONTINUE                                                          00012267
C                                                                       00012269
C   FIND TOTAL AMOUNT OF "A" IN ALL EFFLUENT STREAMS                    00012270
C                                                                       00012271
      FLOAO=0.                                                          00012272
      DO 18 I=1,KO                                                      00012273
         DO 19 J=1,9                                                    00012274
            IF(INT(STRMID(OS(I),J,1)) .EQ. KEYPOS(1)) THEN              00012276
               FLOAO=FLOAO+STRMID(OS(I),J,2)                            00012277
               GOTO 18                                                  00012278
            END IF                                                      00012279
19       CONTINUE                                                       00012280
18    CONTINUE                                                          00012281
      ARCTED=FLOAI-FLOAO                                                00012284
C                                                                       00012285
C   THE YIELD SUBROUTINE ASSUMES THAT THERE IS NO INLET FLOW FOR        00012286
C   THE COMPONENTS THAT YIELD DATA IS DESIRED.                          00012287
C                                                                       00012288
C   SEARCH THE PROCESS OUTLET STREAMS FOR COMPONENTS FOR WHICH          00012289
C   YIELD DATA IS DESIRED.                                              00012290
C                                                                       00012291
      KCMP=1                                                            00012293
24    DO 21 I=1,KO                                                      00012294
         DO 22 J=1,9                                                    00012295
            IF(STRMID(OS(I),J,1) .LE. 0.) GOTO 21                       00012296
            IF(INT(STRMID(OS(I),J,1)) .EQ. YLDCMP(KCMP)) THEN           00012297
               YCMPFL(KCMP)=YCMPFL(KCMP)+STRMID(OS(I),J,2)              00012299
               GOTO 21                                                  00012300
            END IF                                                      00012301
22       CONTINUE                                                       00012302
21    CONTINUE                                                          00012303
      KCMP=KCMP+1                                                       00012304
      IF(KCMP .EQ. 6) GOTO 23                                           00012305
      IF(YLDCMP(KCMP) .EQ. 0) GOTO 23                                   00012306
      KCMP=KCMP+1                                                       00012308
      GOTO 24                                                           00012309
```

```
C                                                                       00012310
C   NOW HAVE ALL THE OUTLET COMPONENT FLOWS FOR WHICH YIELD DATA IS     00012311
C   DESIRED.                                                            00012312
C                                                                       00012313
23    DO 25 I=1,5                                                       00012317
         FY(I)=YCMPFL(I)/ARCTED                                         00012318
25       FYBOF(I)=YCMPFL(I)/FLOAI                                       00012319
      WRITE(NW,103)                                                     00012320
103   FORMAT(//' ************************   YIELD DATA   ***********',  00012321
     *  '************'//' COMPONENT      FRACTIONAL YIELD      FRACT',  00012322
     *  'IONAL YIELD BASED ON FEED'/'-------------------------------',  00012323
     *  '-------------------------------------')                        00012324
      DO 26 I=1,NUMCMP                                                  00012326
         WRITE(NW,104) CPROPS(8,YLDCMP(I)),CPROPS(9,YLDCMP(I)),FY(I),   00012327
     *   FYBOF(I)                                                       00012328
26    CONTINUE                                                          00012329
104   FORMAT(2X,2(A4),13X,F5.2,23X,F5.2)                                00012330
      RETURN                                                            00012331
      END                                                               00012332
      SUBROUTINE EDIT                                                   00012333
C                                                                       00012340
C***********************************************************************00012400
C                                                                       00012500
C   THIS ROUTINE ALLOWS THE USER TO MAKE CHANGES IN THE SIMULATION      00012600
C   PRIOR TO EXECUTION. PRESENTLY (11/01/85), THE ROUTINE IS CALLED     00012700
C   ONLY AT THE BEGINNING OF PROGRAM EXECUTION.                         00012800
C                                                                       00012900
C***********************************************************************00013000
C                                                                       00013100
      COMMON/INOUT/NW,NR                                                00013200
      COMMON /EDFLAG/NEDFLG(2)                                          00013300
      CHARACTER IDUM,IDUM2*2                                            00013400
11    WRITE(NW,100)                                                     00013500
100   FORMAT(/' COMMAND            CHANGE DESIRED'/'-----------------',  00013600
     *       '-------------------------'/)                             00013700
      WRITE(NW,101)                                                     00013800
101   FORMAT( ' FR            PROD STRM FRACNS FROM SPLITTER'/          00013900
     *        ' XA            SET CONVERSION '/                         00014000
     *        ' RS            RESET ENTIRE PROCESS'/)                   00014100
      READ(NR,109) IDUM2                                                00014200
      IF(IDUM2 .EQ. 'RS') NEDFLG(1)=0                                   00014300
      IF(IDUM2 .EQ. 'RS') RETURN                                        00014400
      IF(IDUM2 .EQ. 'FR') CALL CHNGFR                                   00014500
      IF(IDUM2 .EQ. 'XA') CALL CHNGXA                                   00014600
      IF(IDUM2 .EQ. 'OP') CALL CHNGOP                                   00014700
      WRITE(NW,102)                                                     00014800
102   FORMAT(' FINISHED MAKING PROCESS CHANGES? (Y/N)')                 00014900
      READ(NR,103) IDUM                                                 00015000
103   FORMAT(A1)                                                        00015100
109   FORMAT(A2)                                                        00015200
      IF(IDUM .EQ. 'Y' .OR. IDUM .EQ. ' ') GOTO 10                      00015300
      GOTO 11                                                           00015400
10    RETURN                                                            00015500
      END                                                               00015600
      SUBROUTINE CHNGXA                                                 00015700
C                                                                       00015800
C***********************************************************************00015900
C                                                                       00016000
C   THIS ROUTINE ALLOWS THE USER TO CHANGE THE DESIRED CONVERSION.      00016100
C   CHNGXA IS CALLED BY EDIT.                                           00016200
C                                                                       00016300
C***********************************************************************00016400
C                                                                       00016500
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00016600
     *          ,NMRCTR                                                00016700
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                   00016800
     *        ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS   00016900
     *        ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00017000
     *        ,RXN(5,14)                                                00017100
      COMMON /CHARA/MODULE(5),CHRCHK(39),CHRUID(5),SEP(3)               00017200
      COMMON/INOUT/NW,NR                                                00017300
      CHARACTER IDUM,RXN,SEP,CHRCHK,MODULE*8,CHRUID*2                   00017400
```

```
12    DO 10 I=1,MAXUID                                                  00017500
         IF(NPM(I,2) .GT. 2) THEN                                       00017600
            WRITE(NW,100) I,CHRUID(NPM(I,2)),X(I)                       00017700
            READ(NR,*) X(I)                                             00017800
            CALL RERRLM(.01,.99,X(I))                                   00017900
         END IF                                                         00018000
10    CONTINUE                                                          00018100
100   FORMAT(' FOR UNIT # ',I1,' ',A2,' PRESENT CONVERSION IS ',F4.2/   00018200
     *       ' ENTER NEW CONVERSION.')                                  00018300
      WRITE(NW,101)                                                     00018400
101   FORMAT(' FINISHED CHANGING CONVERSIONS? (Y/N)')                   00018500
      READ(NR,102) IDUM                                                 00018600
      IF(IDUM .EQ. 'Y' .OR. IDUM .EQ. ' ') GOTO 11                      00018700
      GOTO 12                                                           00018800
11    RETURN                                                            00018900
102   FORMAT(A1)                                                        00019000
      END                                                               00019100
      SUBROUTINE CHNGOP                                                 00019200
      COMMON/INOUT/NW,NR                                                00019300
      WRITE(NW,100)                                                     00019400
100   FORMAT(' THIS OPTION IS NOT PRESENTLY AVAILABLE. RETURNING TO',   00019500
     *       /' MAIN PROGRAM.')                                         00019600
      RETURN                                                            00019700
      END                                                               00019800
      SUBROUTINE PRCMAP                                                 00019900
C                                                                       00020000
C***********************************************************************00020100
C                                                                       00020200
C   THE PURPOSE OF THIS ROUTINE IS TO ALLOW THE USER TO CHECK THE       00020300
C   TOPOLOGY OF THE ENTIRE PROCESS MAP INCLUDING HOW THE UNITS ARE      00020400
C   TO BE RUN, BEFORE THE SIMULATION IS TO BE EXECUTED.                 00020500
C                                                                       00020600
C***********************************************************************00020700
C                                                                       00020800
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00020900
     *        ,IPHASE,MAXCMP,INERT(5)                                   00021000
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00021100
     *           ,NMRCTR                                                00021200
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                   00021300
     *        ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS   00021400
     *        ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00021500
     *        ,RXN(5,14)                                                00021600
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00021700
     *             , ITRS,ITRT,MAXITR,CTOL                              00021800
      COMMON /CHARA/MODULE(5),CHRCHK(39),SEP(3),CHRUID(5)               00021900
      COMMON/INOUT/NW,NR                                                00022000
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                      00022100
     *            CMP(100)                                              00022200
      INTEGER FLGSP,FLGERR,UNTFLG                                       00022300
      CHARACTER RXN,CHRUID*2,UID*3,DUM(8),CHRCHK,SEP,MODULE*8           00022400
     *          ,IDUM,IDUM2,ISONON*13                                   00022500
      DOUBLE PRECISION FLONRT                                           00022600
      DIMENSION UID(5),ISONON(2)                                        00022700
      DATA ISONON/'ISOTHERMAL','NONISOTHERMAL'/                         00022800
      FLGSP=0                                                           00022900
      WRITE(NW,100)                                                     00023000
100      FORMAT('    UNIT #     UNIT ID          ASSOCIATED  STREAMS'/ 00023100
     *        '-------------------------------------------------',00023200
     *        '------------')                                           00023300
      DO 10 I=1,MAXUID                                                  00023400
         IF(NPM(I,2) .EQ. 2) FLGSP=1                                    00023500
10    WRITE(NW,101) I,CHRUID(NPM(I,2)),(NPM(I,J),J=3,7)                 00023600
101   FORMAT(6X,I1,10X,A2,9X,5(I2,4X))                                  00023700
      DO 11 I=1,MAXUID                                                  00023800
11       WRITE(NW,103) I,CHRUID(NPM(I,2)),ISONON(NOPCND(I))             00023900
103   FORMAT(/' FOR UNIT #',I2,' -',A2,'  OPERATION IS ',A13)           00024000
      IF(FLGSP .EQ. 1) WRITE(NW,104)                                    00024100
104   FORMAT(/'  SP UNIT #    PROD STRM#    FRACN OF FEED STRM ')      00024200
      DO 12 I=1,MAXUID                                                  00024300
         IF(NPM(I,2) .EQ. 2) THEN                                       00024400
            DO 13 II=4,7                                                00024500
13          IF(NPM(I,II) .LT. 0) WRITE(NW,105) I,IABS(NPM(I,II)),       00024600
```

```
     *       SPFRAC(IABS(NPM(I,II)))                                          00024700
             END IF                                                           00024800
12    CONTINUE                                                                00024900
      DO 14 I=1,MAXUID                                                        00025000
         IF(NPM(I,2) .GT. 2) THEN                                             00025100
            WRITE(NW,106) I,CHRUID(NPM(I,2)),X(I)                             00025200
         END IF                                                               00025300
14    CONTINUE                                                                00025400
106   FORMAT(' FOR UNIT # ',I1,' ',A2,' CONVERSION IS ',F4.2)                 00025500
105   FORMAT(7X,I1,12X,I1,15X,F4.2)                                           00025600
      RETURN                                                                  00025700
      END                                                                     00025800
      SUBROUTINE CHNGFR                                                       00025900
C                                                                             00026000
C*************************************************************************** 00026100
C                                                                             00026200
C   THIS ROUTINE IS USED TO CHANGE THE SPLITTER PARAMETERS THAT               00026300
C   SET THE OUTLET STREAM FRACTIONS. CHNGFR IS CALLED BY EDIT.                00026400
C                                                                             00026500
C*************************************************************************** 00026600
C                                                                             00026700
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)          00026800
     *           ,NMRCTR                                                      00026900
      COMMON/INOUT/NW,NR                                                      00027000
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                            00027100
     *            CMP(100)                                                    00027200
      CHARACTER IDUM                                                          00027300
17    N=0                                                                     00027400
      SUMFRC=0.                                                               00027500
      WRITE(NW,100)                                                           00027600
100   FORMAT(' ENTER SP UNIT #')                                              00027700
11    READ(NR,*) NUMSP                                                        00027800
      CALL IERRLM(1,5,NUMSP)                                                  00027900
      DO 9 I=1,MAXUID                                                         00028000
9        IF(NPM(I,2) .EQ. 2 .AND. NPM(I,2) .EQ. NUMSP) N=1                    00028100
      IF(N .EQ. 0) THEN                                                       00028200
         WRITE(NW,102) NUMSP                                                  00028300
102      FORMAT('*****  ERROR  ****    THERE IS NO SP - UNIT # ',I2/          00028400
     *          ' REENTER SP UNIT #')                                        00028500
         GOTO 11                                                              00028600
      END IF                                                                  00028700
14    DO 10 I=4,6                                                             00028800
         IF(NPM(NUMSP,I+1) .EQ. 0) THEN                                       00028900
            K=IABS(NPM(NUMSP,I))                                              00029000
            GOTO 18                                                           00029100
         END IF                                                               00029200
         K=IABS(NPM(NUMSP,I))                                                 00029300
         WRITE(NW,101) NUMSP,K,SPFRAC(K)                                      00029400
101   FORMAT(' FOR SP UNIT #',I2,' FRACN OF FEED STRM TO PROD STRM # ',       00029500
     * I2,' IS ',F4.2,'.')                                                    00029600
         WRITE(NW,107) K                                                      00029700
107      FORMAT(/' ENTER NEW FRACN OF FEED STRM TO PROD STRM # ',I2)          00029800
         READ(NR,*) SPFRAC(K)                                                 00029900
         CALL RERRLM(.01,.99,SPFRAC(K))                                       00030000
         SUMFRC=SUMFRC + SPFRAC(K)                                            00030100
10    CONTINUE                                                                00030200
      SPFRAC(IABS(NPM(NUMSP,7)))=1.-SUMFRC                                    00030300
      GOTO 12                                                                 00030400
18    SPFRAC(K)=1.-SUMFRC                                                     00030500
12    DO 13 I=4,7                                                             00030600
         K=IABS(NPM(NUMSP,I))                                                 00030700
         IF(K .EQ. 0) GOTO 108                                                00030800
         WRITE(NW,103) K,SPFRAC(K)                                            00030900
13    CONTINUE                                                                00031000
108   WRITE(NW,104)                                                           00031100
104   FORMAT(' ARE THESE FRACNS OK? (Y/N)')                                   00031200
      READ(NR,105) IDUM                                                       00031300
      IF(IDUM .EQ. 'Y' .OR. IDUM .EQ. ' ') GOTO 15                            00031400
      GOTO 14                                                                 00031500
15    WRITE(NW,106)                                                           00031600
106   FORMAT(' FINISHED CHANGING STRM FRACNS FOR ALL SP''S? (Y/N)')           00031700
      READ(NR,105) IDUM                                                       00031800
```

```
      IF(IDUM .EQ. 'Y' .OR. IDUM .EQ. ' ') GOTO 16                          00031900
      GOTO 17                                                               00032000
16    RETURN                                                                00032100
103   FORMAT(' FOR STRM # ',I2,' FRACN OF FEED STRM IS ',F4.2)              00032200
105   FORMAT(A1)                                                            00032300
      END                                                                   00032400
      SUBROUTINE TOPOLI                                                     00032500
C*************************************************************************00032600
C                                                                           00032700
C  THE PURPOSE OF THIS SUBROUTINE IS TO ACCEPT DATA FROM THE USER THAT      00032800
C  SETS THE TOPOLOGY OF THE SIMULATION.                                     00032900
C                                                                           00033000
C*************************************************************************00033100
C                                                                           00033200
C  VARIABLES:                                                               00033300
C                                                                           00033400
C         NW    WRITE UNIT NUMBER                                         00033500
C         NR    READ UNIT NUMBER                                          00033600
C        NPM    PROCESS MATRIX. HOLDS UNIT ID #, UNIT TYPE, AND ASSOCIATE 00033700
C               STREAM NUMBERS FOR EACH UNIT.                             00033800
C       NSCM    STREAM CONNECTION MATRIX. HOLDS STREAM #, "FROM" UNIT AND 00033900
C               "TO" UNIT INFORMATION.                                    00034000
C     NSTRMS    NUMBER OF STREAMS IN THE SIMULATION.                      00034100
C     MAXUID    NUMBER OF UNITS IN THE SIMULATION. UNIT ID NUMBERS MUST   00034200
C               START AT 1. THE UNITS ARE THEN NUMBERED SEQUENTIALLY.     00034300
C       NUID    UNIT ID #. 1) STREAM MIXER  2) STREAM SPLITTER  3) PFR    00034400
C               4) CSTR  5) BATCH                                         00034500
C     NOPCND    REACTOR(S) OPERATING CONDITION. 1)ISOTHERMAL              00034600
C               2) NONISOTHERMAL (5).                                     00034700
C     FLGERR    ERROR FLAG.                                               00034800
C                                                                           00034900
C*************************************************************************00035000
C                                                                           00035100
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)        00035200
     *           ,NMRCTR                                                   00035300
      COMMON /INOUT/NW,NR                                                   00035400
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)            00035500
     *             , ITRS,ITRT,MAXITR,CTOL                                00035600
      COMMON /CHARA/MODULE(5),CHRCHK(39),SEP(3),CHRUID(5)                   00035700
      CHARACTER CHRUID*2,UID*3,DUM(8),CHRCHK,SEP,MODULE*8                   00035800
     *             ,IDUM,UNTCHR*20                                          00035900
      INTEGER FLGERR,UNTFLG                                                 00036000
      DIMENSION NOUT(5),UID(5),UNTCHR(5)                                    00036100
      DATA (UNTCHR(I),I=1,5)/'MIXER','SPLITTER','PLUG FLOW RCTR',           00036200
     *                       'STIRRED TANK RCTR','BATCH RCTR'/              00036300
22    WRITE(NW,999)                                                         00036400
999   FORMAT(' PROCESS TOPOLOGY WILL NOW BE SET UP.')                       00036500
      WRITE(NW,100)                                                         00036600
100   FORMAT(' ENTER THE TOTAL NUMBER OF UNITS THAT WILL BE USED IN'        00036700
     */      '       THE SIMULATION.')                                      00036800
      READ(NR,*) MAXUID                                                     00036900
      CALL IERRLM(1,5,MAXUID)                                               00037000
      MAXITR=MAXUID*20                                                      00037100
      WRITE(NW,1000)                                                        00037200
 1000 FORMAT(/' UNIT ID''S WILL NOW BE ENTERED.'/' ENTER UNIT ID''S',       00037300
     *       ' IN THE DESIRED CALCULATIONAL ORDER.'//)                      00037400
      WRITE(NW,102)                                                         00037500
 102  FORMAT(' UNITS AVAILABLE ARE:              UNIT         UNIT ID'/     00037600
     *       '                              ----------------------------'/00037700
     *       '                                 STREAM MIXER       MX'/     00037800
     *       '                               STREAM SPLITTER      SP'/     00037900
     *       '                             PLUG FLOW REACTOR      PF'/     00038000
     *       ' (NOT AVAILABLE)         STIRRED TANK REACTOR       CS'/     00038100
     *       ' (NOT AVAILABLE)               BATCH REACTOR        BR'/)    00038200
      DO 11 I=1,MAXUID                                                      00038300
          WRITE(NW,104) I                                                   00038400
104       FORMAT(' ENTER UNIT ID FOR UNIT ',I2)                             00038500
          READ(NR,109) UID(I)                                               00038600
109       FORMAT(A3)                                                        00038700
20        J=1                                                               00038800
21           IF(CHRUID(J) .EQ. UID(I)) THEN                                 00038900
                NUID(I)=J                                                   00039000
```

```
                GOTO 11                                                 00039100
              ELSE                                                      00039200
                IF(J .EQ. 5) THEN                                       00039300
                  WRITE(NW,108) UID(I),I                                00039400
                  READ(NR,109) UID(I)                                   00039500
                  GOTO 20                                               00039600
                ELSE                                                    00039700
                  J=J+1                                                 00039800
                  GOTO 21                                               00039900
                END IF                                                  00040000
              END IF                                                    00040100
C           READ(NR,*) NUID(I)                                          00040200
C           CALL IERRLM(1,5,NUID(I))                                    00040300
            NPM(I,1)=I                                                  00040400
            NPM(I,2)=NUID(I)                                            00040500
11      CONTINUE                                                        00040600
108     FORMAT(1X,A3,' CHARACTERS ARE NOT ACCEPTABLE UNIT ID''S'/       00040700
     *                        ' REENTER UNIT ID FOR UNIT',I2)           00040800
C                                                                       00040900
C     KNT1 TRACKS COL # IN PROCESS MATRIX NPM.                          00041000
C     I TRACKS THE UNIT # AND ROW # IN PROCESS MATRIX NPM.              00041100
C     JI TRACKS CHARACTER # IN DUM.                                     00041200
C                                                                       00041300
      WRITE(NW,113)                                                     00041400
      DO 10 I=1,MAXUID                                                  00041500
C                                                                       00041600
C     SET PROCESS MATRIX UNIT NUMBERS (I,1) AND PROCESS MATRIX          00041700
C     ID #'S (I,2).                                                     00041800
C                                                                       00041900
      NPM(I,1)=I                                                        00042000
      NPM(I,2)=NUID(I)                                                  00042100
      KNT1=3                                                            00042200
C                                                                       00042300
C     NSTUNT=# OF STREAMS PER UNIT                                      00042400
C                                                                       00042500
      NSTUNT=0                                                          00042600
      ISIGN=1                                                           00042700
113   FORMAT(' STREAM IDENTIFICATION NUMBERS WILL NOW BE ENTERED.'/     00042800
     *       ' THE MAXIMUM ALLOWABLE STREAM ID NUMBER IS 9.'            00042900
     *      /' THERE IS NO CHECK ON THIS VARIABLE.'//)                  00043000
26    WRITE(NW,101) I,UNTCHR(NUID(I))                                   00043100
101   FORMAT(' ENTER STREAM NUMBERS FOR ALL STREAMS ENTERING UNIT',I2,  00043200
     *       '- ',A17)                                                  00043300
      READ(NR,110) (DUM(JI),JI=1,8)                                     00043400
      IF(DUM(1) .EQ. ' ') THEN                                          00043500
         WRITE(NW,112)                                                  00043600
112      FORMAT(' *****  ERROR  *****'/' A BLANK WAS ENTERED AS THE ',  00043700
     *   'FIRST CHARACTER'/' REENTER INPUT LINE.')                      00043800
         GOTO 26                                                        00043900
      END IF                                                            00044000
110   FORMAT(8A1)                                                       00044100
25    DO 23 JI=1,8                                                      00044200
         DO 24 JII= 27,35                                               00044300
            IF(DUM(JI) .EQ. CHRCHK(JII)) THEN                           00044400
               NSTUNT=NSTUNT+1                                          00044500
               NPM(I,KNT1)=ISIGN*(JII-26)                               00045100
               KNT1=KNT1+1                                              00045200
               GOTO 23                                                  00045300
            END IF                                                      00045400
24       CONTINUE                                                       00045500
23    CONTINUE                                                          00045600
      IF(ISIGN .EQ. -1) GOTO 10                                         00045700
      WRITE(NW,111) I,UNTCHR(NUID(I))                                   00045800
111   FORMAT(' ENTER STREAM NUMBERS FOR ALL STREAMS LEAVING UNIT',I2,   00045900
     *       '- ',A17)                                                  00046000
      READ(NR,110) (DUM(JI),JI=1,8)                                     00046100
      ISIGN=-1                                                          00046200
      GOTO 25                                                           00046300
10    CONTINUE                                                          00046400
      CALL IOCHK                                                        00046500
      IF(FLGERR .EQ. 1) THEN                                            00046600
         GOTO 22                                                        00046700
```

```
      END IF                                                                    00046800
      WRITE(NW,103)                                                             00046900
103   FORMAT(' DO YOU WANT TO CHECK THE PROCESS TOPOLOGY? (Y/N)')               00047000
      READ(NR,105) IDUM                                                         00047100
105   FORMAT(A1)                                                                00047200
      IF(IDUM .EQ. 'Y' .OR. IDUM .EQ. ' ') THEN                                 00047300
         WRITE(NW,106)                                                          00047400
106      FORMAT('    UNIT #     UNIT ID          ASSOCIATED  STREAMS'/         00047500
     *          '-------------------------------------------------',00047600
     *          '------------')                                                 00047700
         DO 13 I=1,MAXUID                                                       00047800
13       WRITE(NW,107) I,CHRUID(NPM(I,2)),(NPM(I,J),J=3,7)                      00047900
107      FORMAT(6X,I1,10X,A2,8X,5(I2,4X))                                       00048000
         WRITE(NW,115)                                                          00048100
115      FORMAT(/' IS EVERYTHING OK? (Y/N)')                                    00048200
         READ(NR,105) IDUM                                                      00048300
         IF(IDUM .EQ. 'Y' .OR. IDUM .EQ. ' ') GOTO 14                           00048400
         WRITE(NW,114)                                                          00048500
114      FORMAT(' PROCESS TOPOLOGY WILL HAVE TO BE RESET.')                     00048600
         CALL INISHL                                                            00048700
         GOTO 22                                                                00048800
      ELSE                                                                      00048900
         GOTO 14                                                                00049000
      END IF                                                                    00049100
C                                                                               00049200
C   FIND THE NUMBER OF REACTORS IN THE PROCESS.                                 00049300
C                                                                               00049400
14    DO 12 I=1,9                                                               00049500
         IF(NPM(I,2) .EQ. 3 .OR. NPM(I,2) .EQ. 4 .OR. NPM(I,2) .EQ. 5)          00049600
     *   NMRCTR=NMRCTR + 1                                                      00049700
12    CONTINUE                                                                  00049800
      RETURN                                                                    00050200
      END                                                                       00050300
      SUBROUTINE INISHL                                                         00050400
C*************************************************************************** 00050500
C                                                                               00050600
C   THE PURPOSE OF THIS ROUTINE IS TO INITIALIZE PROGRAM VARIABLES              00050700
C   THIS ROUTINE IS CALLED AT THE BEGINNING OF PROGRAM EXECUTION                00050800
C   AND WHENEVER A NEW SIMULATION IS RUN.                                       00050900
C                                                                               00051000
C*************************************************************************** 00051100
C                                                                               00051200
C   VARIABLES:                                                                  00051300
C                                                                               00051400
C       UNTFLG    (INDX,1)=0 IF UNIT HAS NOT BEEN CALCULATED AT                 00051500
C                 LEAST ONCE.                                                   00051600
C                 (INDX,1)=1 IF UNIT HAS BEEN CALC'D ATLEAST ONCE.              00051700
C                 (INDX,2)=0 IF WANT UNIT TO BE CALC'D.                         00051800
C                 (INDX,2)=1 IF DON'T WANT UNIT TO BE CALC'D.                   00051900
C       FLGERR    =1 FOR ALL IS OK                                              00052000
C                 =2 IF AN ERROR HAS BEEN FOUND                                 00052100
C                                                                               00052200
C      NFLGCV    (INDX)=0 IF THERE ARE NO RECYCLE STREAMS IN THE PROCESS       00052300
C                         OR A UNIT HAS CONVERGED.                              00052400
C                                                                               00052500
C                       =1 IF THERE ARE RECYCLE STREAMS IN THE PROCESS.         00052600
C                                                                               00052700
C       NFLAG(INDX)    =  IS A COUNTER IN ODE THAT WILL TELL HOW MANY           00052800
C                         TIMES THAT ODE IS CALLED TO INTEGRATE FOR             00052900
C                         A GIVEN REACTOR (DISTINGUISHED BY INDX). IF           00053000
C                         IF NFLAG(INDX) IS > 1 THEN THERE MUST BE              00053100
C                         RECYCLE STREAM(S) PRESENT. THIS WILL BE USED          00053200
C                         TO TELL FEX WHEN TO SET NCOUNT = 3 AND FILL           00053300
C                         THE RCT ARRAY SEQUENTIALLY. IF NFLAG(INDX) IS         00053310
C                         > 1 AND V(VOLUME) OR T IS =0.0 THEN FEX WILL          00053320
C                         KNOW THAT ODE IS BEING CALLED AGAIN AND MUST          00053330
C                         SET NCOUNT(WHICH INCREMENTS THE RCT ARRAY) =3.         00053340
C                                                                               00053350
C                                                                               00053400
C           E    ARRAY THAT HOLDS THE VALUES FOR ACTIVATION ENERGY.             00053500
C                E(1,1)=E FOR FORWARD REACTION #1.                              00053600
C                E(1,2)=E FOR REVERSE REACTION #1.                              00053700
```

```
C                                                                       00053800
C       NEDFLG(1) = 1 = EVERYTHING OK IN CALL TO EDIT. I.E. THERE       00053900
C                         ARE NO CHANGES TO BE MADE.                    00054000
C                                                                       00054100
C***********************************************************************00054200
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00054300
     *        ,IPHASE,MAXCMP,INERT(5)                                   00054400
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00054500
     *        ,NMRCTR                                                   00054600
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                   00054700
     *        ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS   00054800
     *        ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00054900
     *        ,RXN(5,14)                                                00055000
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                      00055100
     *          CMP(100)                                                00055200
      COMMON /INOUT/NW,NR                                               00055400
      COMMON /EDFLAG/NEDFLG(2)                                          00055500
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00055600
     *             , ITRS,ITRT,MAXITR,CTOL                              00055700
      DOUBLE PRECISION FLONRT                                           00055800
      CHARACTER RXN,CHRUID*2,CHRCHK,SEP,MODULE*8                        00055900
      INTEGER FLGERR,UNTFLG                                             00056000
      NW=6                                                              00056100
      NR=9                                                              00056200
      NSTRMS=0                                                          00056300
      MAXUID=0                                                          00056400
      MAXCMP=9                                                          00056500
      NFLAG1=0                                                          00056600
      FLGERR=0                                                          00056700
      NRXNS=0                                                           00056800
      NRXNTS=0                                                          00056900
      KEY=0                                                             00057000
      FAO=-1.                                                           00057100
      CTOL=.00001                                                       00057200
      NMRCTR=0                                                          00057400
      ITRS=0                                                            00057500
      ITRT=0                                                            00057600
      NEDFLG(1)=1                                                       00057700
      NEDFLG(2)=1                                                       00057800
      DO 10 I=1,9                                                       00057900
         DO 11 J=1,7                                                    00058000
            NPM(I,J)=0                                                  00058100
            DO 88 K=1,5                                                 00058200
88          NPOSPC(J,K)=0                                               00058300
11       CONTINUE                                                       00058400
10    CONTINUE                                                          00058500
      DO 12 I=1,9                                                       00058600
         SPFRAC(I)=0.                                                   00058700
         DO 13 J=1,3                                                    00058800
13          NSCM(I,J)=0                                                 00058900
12    CONTINUE                                                          00059000
      DO 15 I=1,9                                                       00059100
         DO 87 K=1,5                                                    00059200
87          EXP(I,K)=0.                                                 00059300
         DO 14 J=1,6                                                    00059400
14       NCOEF(I,J)=0                                                   00059500
15    CONTINUE                                                          00059600
      DO 22 I=1,5                                                       00059700
         NFLAG(I)=0                                                     00059710
         RCTVOL(I)=0.                                                   00059800
         NRXTYP(I)=0                                                    00059900
         INERT(I)=0                                                     00060000
         KEYPOS(I)=0                                                    00060100
         DO 16 J=1,2                                                    00060200
            A(J)=0.                                                     00060300
            E(J)=0.                                                     00060400
16       RK(I,J)=0.                                                     00060500
         IDIR(I)=0                                                      00060600
         NOPCND(I)=0                                                    00060700
         NFLGCV(I)=0                                                    00060900
         UNTFLG(I,1)=0                                                  00061000
22       UNTFLG(I,2)=0                                                  00061100
```

```
      DO 17 I=1,9                                                       00061200
         DO 18 J=1,9                                                    00061300
18       STFLO(I,J)=0.                                                  00061400
17    CONTINUE                                                          00061500
      DO 23 I=1,9                                                       00061600
         DO 24 J=1,10                                                   00061700
            DO 25 K=1,7                                                 00061800
25          STRMID(I,J,K)=-1.                                           00061900
24       CONTINUE                                                       00062000
23    CONTINUE                                                          00062100
      DO 33 K=1,5                                                       00062200
         DO 31 I=1,100                                                  00062300
            DO 32 J=1,14                                                00062400
32             RCT(K,I,J)=0.                                            00062500
31       CONTINUE                                                       00062600
33    CONTINUE                                                          00062700
      DO 34 I=1,6                                                       00062800
34       FLONRT(I)=0.DO                                                 00062900
      RETURN                                                            00063000
      END                                                               00063100
      SUBROUTINE STRMI                                                  00063200
C                                                                       00063300
C*********************************************************************  00063400
C                                                                       00063500
C   THE PURPOSE OF THIS ROUTINE IS TO ACCEPT STREAM DATA FROM THE       00063600
C   USER ( ID #'S, FLOWRATES FOR PLUG FLOW OR STIRRED TANK REACTORS     00063700
C   THIS ROUTINE ALSO REQUESTS PROCESS FEED STREAM TEMPERATURES IF      00063800
C   IPHASE = 1 I.E. GAS.                                                00063900
C                                                                       00064000
C*********************************************************************  00064100
C                                                                       00064200
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00064300
     *        ,IPHASE,MAXCMP,INERT(5)                                   00064400
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00064500
     *          ,NMRCTR                                                 00064600
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00064700
     *       , ITRS,ITRT,MAXITR,CTOL                                    00064800
      COMMON /INOUT/NW,NR                                               00064900
      DOUBLE PRECISION FLONRT                                           00065000
      INTEGER FLGERR,UNTFLG                                             00065100
19    I=1                                                               00065500
      WRITE(NW,100)                                                     00065600
100   FORMAT(' ENTER FLUID PHASE.    1) GAS      2) LIQUID (NOT AVAIL',  00065700
     *       'ABLE)'/)                                                  00065800
      READ(NR,*) IPHASE                                                 00065900
      CALL IERRLM(1,1,IPHASE)                                           00066000
      IF(IPHASE .EQ. 1) THEN                                            00066100
         WRITE(NW,102)                                                  00066200
         READ(NR,*) P                                                   00066300
         CALL RERRLM(.1E-5,5000.,P)                                     00066700
      END IF                                                            00066800
102   FORMAT(' ENTER SYSTEM PRESSURE (ATM)')                            00066900
      DO 12 I=1,MAXUID                                                  00067000
         IF(NUID(I) .EQ. 5) NFLAG1=1                                    00067100
12    CONTINUE                                                          00067200
      IF(NFLAG1 .EQ. 1) GOTO 13                                         00067300
      WRITE(NW,103)                                                     00067400
103   FORMAT(/' PROCESS FEED STREAM DATA WILL NOW BE ENTERED.'/         00067500
     *     ' THERE IS A MAXIMUM OF NINE COMPONENTS PER STREAM'/         00067600
     *     ' AND NINE DIFFERENT COMPONENTS IN THE PROCESS.'//)          00067700
      DO 44 I=1,NSTRMS                                                  00067800
         IF( NSCM(I,2) .EQ. 0) THEN                                     00067900
           IF(IPHASE .EQ. 1) GOTO 15                                    00068000
           WRITE(NW,106) I                                              00068100
106        FORMAT(' ENTER VOLUMETRIC FLOW RATE (M**3/MIN) FOR',         00068200
     *     ' STREAM ',I1)                                               00068300
           READ(NR,*) STRMID(I,10,6)                                    00068400
           CALL RERRLM (1.E-6,1.E6,STRMID(I,10,6))                      00068500
           GOTO 16                                                      00068600
15          WRITE(NW,104) I                                             00068700
            READ(NR,*) STRMID(I,10,1)                                   00068800
            STRMID(I,10,1)=STRMID(I,10,1) + 273.15                      00068900
```

```
          WRITE(NW,101) I                                                    00069000
101       FORMAT(' ENTER ID NUMBERS AND FLOWRATES (KG-MOLES/MIN)',           00069100
     *          ' FOR COMPONENTS IN STREAM ',I1,'.'/' ENTER 0,0 TO ',       00069200
     *           'TERMINATE INPUT FOR EACH STREAM.')                        00069300
104       FORMAT(' ENTER TEMPERATURE (C) FOR PROCESS FEED STREAM',I4/)       00069400
          GOTO 17                                                            00069500
16        WRITE(NW,107) I                                                    00069600
107       FORMAT(' ENTER ID NUMBERS AND INITIAL CONCENTRATIONS ',            00069700
     *    '(KMOL/M**3)'/' FOR COMPONENTS IN STREAM',I2,'.'/' ENTER',         00069800
     *    ' 0,0 TO TERMINATE INPUT FOR EACH STREAM.')                        00069900
17        DO 10 K=1,MAXCMP                                                   00070000
             FLGERR=1                                                        00070100
C                                                                            00070200
C    FLGERR = 1 INPUT DATA OK.                                               00070300
C                                                                            00070400
             READ(NR,*) STRMID(I,K,1),STRMID(I,K,2)                          00070500
          IF(INT(STRMID(I,K,1)) .EQ. 0 .AND. INT(STRMID(I,K,2)) .EQ. 0)      00070600
     *       THEN                                                            00070700
                STRMID(I,K,1)=-1.                                            00070800
                STRMID(I,K,2)=-1.                                            00070900
                GOTO 44                                                      00071000
             END IF                                                          00071100
20           IF(STRMID(I,K,2) .LE. 0.) THEN                                  00071200
                FLGERR=0                                                     00071300
                WRITE(NW,203) INT(STRMID(I,K,1))                             00071400
203          FORMAT(' FLOWRATES CANNOT BE LESS THAN OR EQUAL TO 0.'/         00071500
     *          ' PLEASE REENTER FLOWRATE FOR CMP #',I3)                     00071600
             READ(NR,*) STRMID(I,K,2)                                        00071700
             GOTO 20                                                         00071800
             END IF                                                          00071900
201       IF(INT(STRMID(I,K,1)) .GT. 61 .OR.INT(STRMID(I,K,1)) .LT. 1)       00072000
     *    THEN                                                              00072100
             FLGERR=0                                                        00072200
             IF(STRMID(I,K,1) .GT. 0.) THEN                                  00072300
                WRITE(NW,200)                                                00072400
200             FORMAT(' MAX COMPONENT ID # IS 61. PLEASE REENTER',          00072500
     *                 ' COMPONENT ID #')                                    00072600
                READ(NR,*) STRMID(I,K,1)                                     00072700
                GOTO 201                                                     00072800
             END IF                                                          00072900
             WRITE(NW,202)                                                   00073000
202          FORMAT(' MINIMUM COMPONENT ID # IS 1. PLEASE REENTER',          00073100
     *                 ' COMPONENT ID #')                                    00073200
             READ(NR,*) STRMID(I,K,1)                                        00073300
             GOTO 201                                                        00073400
          END IF                                                             00073500
          IF(FLGERR .EQ. 0) WRITE(NW,108) I                                  00073600
108       FORMAT(' CONTINUE TO ENTER DATA FOR STREAM',I2,'.')                00073700
10      CONTINUE                                                           00073800
        END IF                                                             00074800
44    CONTINUE                                                             00074900
C                                                                            00075000
C    GOTO AROUND BATCH INPUT                                                 00075100
C                                                                            00075200
      GOTO 14                                                                00075300
C                                                                            00075400
C    NOW HAVE ALL COMPONENT ID #'S FOR FEED STREAMS IN THE STRMID ARRAY.     00075500
C                                                                            00075600
13    WRITE(NW,105) I                                                        00075700
105   FORMAT( ' ENTER COMPONENT ID # AND INITIAL CHARGE (MOLES) FOR ',       00075800
     *   'COMPONENT',I1/)                                                    00075900
      READ(NR,*) STRMID(1,I,1),STRMID(1,I,2)                                 00076000
      IF(STRMID(1,I,1) .EQ. 0.) GOTO 14                                      00076100
      I=I+1                                                                  00076200
      GOTO 13                                                                00076300
 14   RETURN                                                                 00077100
      END                                                                    00077200
      SUBROUTINE RERRLM(LOW,HIGH,CHEK)                                       00077300
C**********************************************************************      00077400
C                                                                            00077500
C    THE PURPOSE OF THIS ROUTINE IS TO CHECK REAL INPUT DATA THAT IS         00077600
C    OUT OF RANGE. THE ARGUMENTS OF THE SUBROUTINE ARE LOWEST AC-            00077700
```

```
C   CEPTABLE VALUE, HIGHEST ACCEPTABLE VALUE, AND THE VARIABLE          00077800
C   TO BE CHECKED. FLAGERR IS SET TO O IF THERE IS NO ERROR             00077900
C   OR IS SET TO 1 IF THERE IS AN ERROR. IF THE VALUE IS OUT OF         00078000
C   RANGE, AN NEW VALUE IS REQUESTED FROM THE USER, CHECKED FOR         00078100
C   ACCEPTABILITY, AND RETURNED TO THE CALLING PROGRAM IF OK.           00078200
C                                                                       00078300
C*********************************************************************  00078400
      REAL LOW                                                          00078500
      COMMON /INOUT/NW,NR                                               00078600
10    IF(CHEK .GT. HIGH .OR. CHEK .LT. LOW) THEN                        00078700
        IF(CHEK .GT. HIGH) THEN                                         00078800
          WRITE(NW,100) HIGH                                            00078900
100       FORMAT(' INPUT ERROR. MAXIMUM INPUT VALUE IS ',F10.3,'.',     00079000
     *      ' REENTER INPUT LINE.'/)                                    00079100
          READ(NR,*) CHEK                                               00079200
          GOTO 10                                                       00079300
        ELSE                                                            00079400
          WRITE(NW,101) LOW                                             00079500
101       FORMAT(' INPUT ERROR. MINIMUM VALUE IS ',F 10.3,'. REENTER ', 00079600
     *      'INPUT LINE.'/)                                             00079700
          READ(NR,*) CHEK                                               00079800
          GOTO 10                                                       00079900
        END IF                                                          00080000
      ELSE                                                              00080100
      RETURN                                                            00080200
      END IF                                                            00080300
      RETURN                                                            00080400
      END                                                               00080500
      SUBROUTINE IERRLM(NLOW,NHIGH,NCHEK)                               00080900
C*********************************************************************  00081000
C                                                                       00081100
C   THE PURPOSE OF THIS ROUTINE IS TO CHECK REAL INPUT DATA THAT IS     00081200
C   OUT OF RANGE. THE ARGUMENTS OF THE SUBROUTINE ARE LOWEST AC-        00081300
C   CEPTABLE VALUE, HIGHEST ACCEPTABLE VALUE, AND THE VARIABLE          00081400
C   TO BE CHECKED. FLAGERR IS SET TO O IF THERE IS NO ERROR             00081500
C   OR IS SET TO 1 IF THERE IS AN ERROR. IF THE VALUE IS OUT OF         00081600
C   RANGE, AN NEW VALUE IS REQUESTED FROM THE USER, CHECKED FOR         00081700
C   ACCEPTABILITY, AND RETURNED TO THE CALLING PROGRAM IF OK.           00081800
C                                                                       00081900
C*********************************************************************  00082000
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00082100
     *            , ITRS,ITRT,MAXITR,CTOL                               00082200
      COMMON /INOUT/NW,NR                                               00082300
      INTEGER FLGERR,UNTFLG                                             00082400
10    IF(NCHEK .GT. NHIGH .OR. NCHEK .LT. NLOW) THEN                    00082500
        FLGERR=1                                                        00082600
        IF(NCHEK .GT. NHIGH) THEN                                       00082700
          WRITE(NW,100) NHIGH                                           00082800
100       FORMAT(' INPUT ERROR. MAXIMUM INPUT VALUE IS ',I3,'.',        00082900
     *      ' REENTER INPUT LINE.'/)                                    00083000
          READ(NR,*) NCHEK                                              00083100
          GOTO 10                                                       00083200
        ELSE                                                            00083300
          WRITE(NW,101) NLOW                                            00083400
101       FORMAT(' INPUT ERROR. MINIMUM VALUE IS ',I3,'. REENTER ',     00083500
     *      'INPUT LINE.'/)                                             00083600
          READ(NR,*) NCHEK                                              00083700
          GOTO 10                                                       00083800
        END IF                                                          00083900
      ELSE                                                              00084000
      FLGERR=0                                                          00084100
      RETURN                                                            00084200
      END IF                                                            00084300
      RETURN                                                            00084400
      END                                                               00084500
      SUBROUTINE RXNI                                                   00084600
C                                                                       00084700
C*********************************************************************  00084800
C                                                                       00084900
C   THE PURPOSE OF THIS ROUTINE IS TO ACCEPT INFORMATION CONCERNING     00085000
C   THE REACTION TO BE MODELLED. THE REACTION IS INPUT AS CHARACTER     00085100
C   DATA. FROM THIS INFORMATION, RXNI WILL EXTRACT NUMBER OF REACTIONS  00085200
```

```
C   AND REACTANT IDENTIFICATION NUMBERS.                                    00085300
C                                                                           00085400
C***********************************************************************  00085500
C                                                                           00085600
C   VARIABLES:                                                              00085700
C                                                                           00085800
C        RXN   CHARACTER ARRAY THAT HOLDS EACH REACTION. DIMENSIONED        00085900
C              TO (MAX # REVERSIBLE REACTIONS, MAX # CHARACTERS PER         00086000
C              REACTION).                                                   00086100
C        SEP   CHARACTER ARRAY THAT HOLDS "SEPARATORS". SEPARATORS          00086200
C              ARE DEFINED AS "+","=",">". THESE CHARACTERS SEPAR-          00086300
C              ATE REACTANTS AND PRODUCTS IN THE RXN ARRAY.                 00086400
C     NRXTYP   REACTION TYPE. 1) ELEMENTARY   2) NON-ELEMENTARY             00086500
C     NRXNTS   NUMBER OF REACTANTS AND PRODUCTS FOR THE SET                 00086600
C              OF REACTINS TO BE MODELLED. ALTERNATELY, NRXNTS              00086700
C              IS THE NUMBER OF NON ZERO ROWS (1ST COLUMN) IN NCOEF.        00086800
C     NRXCID   REACTANT COMPONENT ID #. DIMENSIONED TO MAX # REACT-         00086900
C              ANTS AND PRODUCTS.                                           00087000
C      NCOEF   HOLDS(SPECIES,REACTION). ENTRIES ARE THE STOICHIOMETRIC      00087100
C              COEFFICIENTS OF THE REACTANTS AND PRODUCTS. THIS ARRAY        00087200
C              IS INITIALIZED TO 0.                                         00087300
C      ISIGN   SIGN (+ FOR REACTANTS, - FOR PRODUCTS) THAT THE              00087400
C              STOICHIOMETRIC COEFFICIENT IN NCOEF HAS.                     00087500
C       IDIR   KEEPS TRACK OF WHETHER A REACTION IS REVERSIBLE (=1)         00087600
C              OR IRREVERSIBLE (=0). (6)                                    00087700
C                                                                           00087800
C        DUM  A CHARACTER VARIABLE THAT ALLOWS THE USER TO HIT "RETURN"     00087900
C              INSTEAD OF INPUTTING A NUMBER.                               00088000
C                                                                           00088100
C      NMRCTR IS EQUAL TO THE NUMBER OF REACTORS IN THE PROCESS.            00088200
C                                                                           00088400
C***********************************************************************  00088500
C                                                                           00088600
      COMMON /CHARA/MODULE(5),CHRCHK(39),SEP(3),CHRUID(5)                   00088700
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)        00088800
     *           ,NMRCTR                                                    00088900
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                       00089000
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS      00089100
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)          00089200
     *         ,RXN(5,14)                                                   00089300
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)            00089400
     *             , ITRS,ITRT,MAXITR,CTOL                                 00089500
      COMMON /INOUT/NW,NR                                                   00089600
      CHARACTER DUM,RXN,SEP,CHRCHK,MODULE*8,CHRUID*2                        00089700
      INTEGER FLGERR,UNTFLG                                                 00089800
      DIMENSION IDUM(26)                                                    00089900
      IF(NMRCTR .EQ. 0) RETURN                                              00090000
      DO 28 I=1,26                                                          00090100
28       IDUM(I)=0                                                          00090200
      I=1                                                                   00090300
      NRCOEF=1                                                              00090400
105   FORMAT(4A1)                                                           00090500
10    WRITE(NW,101)                                                         00090600
101   FORMAT(' KINETIC DATA WILL NOW BE ENTERED.'                           00090700
     *       /' UP TO 14 CHARACTERS PER REACTION CAN BE ACCEPTED.'///)      00090800
22    WRITE(NW,100) I                                                       00090900
100   FORMAT(' ENTER REACTION TYPE FOR REACTION # ',I1,':',                 00091000
     *   5X, ' 1) ELEMENTARY   2) NONELEMENTARY'/)                          00091100
      READ(NR,105) DUM                                                      00091200
      IF(DUM .EQ. ' ' .OR. DUM .EQ. '1') THEN                               00091400
         NRXTYP(I)=1                                                        00091500
         GOTO 23                                                            00091600
      END IF                                                                00091700
      NRXTYP(I)=2                                                           00091800
23    WRITE(NW,104) I                                                       00091900
104   FORMAT(' ENTER REACTION # ',I2,                                       00092000
     *       /' HIT <RETURN> TO TERMINATE DATA ENTRY.')                     00092100
331   READ(NR,102) (RXN(I,J),J=1,14)                                        00092200
      IF(RXN(I,1) .EQ. ' ' .AND. I .EQ. 1) THEN                             00092210
         WRITE(NW,332)                                                      00092220
332      FORMAT(' A BLANK CHARACTER WAS ENTERED FOR A REACTION. PLEASE'     00092230
     *          ,' ENTER'/' A SPECIFIC REACTION.')                          00092240
```

```
         GOTO 331                                                       00092250
       END IF                                                           00092260
       IF(RXN(I,1) .EQ. ' ') GOTO 21                                    00092300
44     ISIGN=-1                                                         00092400
       J=1                                                              00092500
C                                                                       00092600
C    SEARCH FOR A NUMERIC CHARACTER                                     00092700
C                                                                       00092800
47     DO 40 III=27,35                                                  00092900
40        IF(RXN(I,J) .EQ. CHRCHK(III)) GOTO 41                         00093000
83     WRITE(NW,200) I,I                                                00093100
200    FORMAT(' *****  ERROR  *****    CHARACTERS IN REACTION ',I2,     00093200
     *       ' ARE OUT OF SEQUENCE OR AN'//' ILLEGAL CHARACTER HAS BEEN', 00093300
     *       ' INPUT.'//' LEGAL CHARACTERS FOR EACH REACTION MUST BE'/    00093400
     *       ' ENTERED IN THE FOLLOWING SEQUENCE:'/                       00093500
     *       ' STOICHIOMETRIC COEFFICIENT(1,2,..9), SPECIES(A,B,..Z),',   00093600
     *       ' SEPARATOR(+,>,=)'//' PLEASE REENTER REACTION ',I2)         00093700
       READ(NR,102) (RXN(I,J),J=1,14)                                   00093800
       DO 39 M=1,8                                                      00093900
39        NCOEF(M,I+1)=0                                                00094000
       GOTO 44                                                          00094100
41     J=J+1                                                            00094200
       N=(III-26)*ISIGN                                                 00094300
C                                                                       00094400
C    SEARCH FOR AN ALPHABETIC CHARACTER                                 00094500
C                                                                       00094600
       DO 42 III=1,26                                                   00094700
42        IF(RXN(I,J) .EQ. CHRCHK(III)) GOTO 43                         00094800
       GOTO 83                                                          00094900
43     IF(IDUM(III) .EQ. 0) THEN                                        00095000
          IDUM(III)=NRCOEF                                              00095100
          WRITE(NW,201) CHRCHK(III)                                     00095200
201       FORMAT(' ENTER COMPONENT ID # FOR COMPONENT ',A2)             00095300
          READ(NR,*) NCOEF(NRCOEF,1)                                    00095400
          NCOEF(NRCOEF,I+1)=N                                           00096200
          NRCOEF=NRCOEF+1                                               00096300
          GOTO 45                                                       00096400
       ELSE                                                             00096500
          NCOEF(IDUM(III),I+1)=N                                        00096600
          GOTO 45                                                       00096700
       END IF                                                           00096800
45     J=J+1                                                            00096900
       IF(J .EQ. 15) GOTO 48                                            00097000
C                                                                       00097100
C    SEARCH FOR A SEPARATOR                                             00097200
C                                                                       00097300
       IF(RXN(I,J) .EQ. ' ') GOTO 48                                    00097400
       IF(RXN(I,J) .EQ. '+') GOTO 46                                    00097500
       IF(RXN(I,J) .EQ. '>') THEN                                       00097600
          ISIGN=1                                                       00097700
          GOTO 46                                                       00097800
       END IF                                                           00097900
       IF(RXN(I,J) .EQ. '=') THEN                                       00098000
          IDIR(I)=1                                                     00098100
          ISIGN=1                                                       00098200
          GOTO 46                                                       00098300
       END IF                                                           00098400
       GOTO 83                                                          00098500
46     J=J+1                                                            00098600
       GOTO 47                                                          00098700
48     I=I+1                                                            00098800
       IF(I .EQ. 6 ) GOTO 443                                           00098900
102    FORMAT(15A1)                                                     00099000
       GOTO 22                                                          00099100
443    CONTINUE                                                         00099600
 21    CONTINUE                                                         00100000
       NRXNS=I-1                                                        00100200
       CALL FNDKEY                                                      00100500
       NRXNTS=0                                                         00100600
       DO 49 I=1,8                                                      00100700
49        IF(NCOEF(I,1) .NE. 0) NRXNTS=NRXNTS+1                         00100800
        CALL STEXP                                                      00101400
```

```
      DO 50 J=1,5                                                       00101800
         IF(NRXTYP(J) .EQ. 2) THEN                                      00101900
         WRITE(NW,333) J                                                00102000
333      FORMAT(' REACTION #',I2,' HAS BEEN SPECIFIED NONELEMENTARY.'/  00102100
     *       ' PLEASE ENTER THE NEW EXPONENTS IN SEQUENTIAL ORDER.'//)  00102200
            I=1                                                         00102300
            K=1                                                         00102400
53          IF(I .GT. 9) GOTO 52                                        00102500
               IF(NCOEF(I,J+1) .LT. 0) THEN                             00102600
                  WRITE(NW,106) K                                       00102700
106               FORMAT(' ENTER EXPONENT FOR SPECIES ',I1,'.')         00102800
                  READ(NR,*) EXP(I,J)                                   00102900
                  K=K + 1                                               00103000
                  GOTO 51                                               00103100
               ELSE                                                     00103200
                  IF(NCOEF(I,J+1) .EQ. 0) GOTO 51                       00103300
               END IF                                                   00103400
               IF(IDIR(J) .EQ. 1) THEN                                  00103500
                  WRITE(NW,106) K                                       00103600
                  READ(NR,*) EXP(I,J)                                   00103700
                  K=K+1                                                 00103800
               END IF                                                   00103900
51             I=I+1                                                    00104000
            GOTO 53                                                     00104100
52          CONTINUE                                                    00104200
         END IF                                                         00104300
50    CONTINUE                                                          00104400
      RETURN                                                            00104900
      END                                                               00105000
      SUBROUTINE MXMNIO                                                 00107200
C*********************************************************************** 00107300
C                                                                       00107400
C   THIS ROUTINE WILL FIND ALL INLET AND OUTLET STREAMS.                00107500
C   IT ALSO CHECKS TO SEE THAT NO MORE THAN SIX INLET AND OUTLET        00107600
C   STREAMS EXIST FOR THE GIVEN PROCESS.                                00107700
C                                                                       00107800
C*********************************************************************** 00107900
C                                                                       00108000
C   VARIABLES:                                                          00108100
C                                                                       00108200
C      OUTPTS   HOLDS THE NUMBER OF OUTPUT STREAMS.                     00108300
C       INPTS   HOLDS THE NUMBER OF INPUT STREAMS.                      00108400
C                                                                       00108500
C*********************************************************************** 00108600
C                                                                       00108700
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00108800
     *         ,IPHASE,MAXCMP,INERT(5)                                  00108900
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00109000
     *           ,NMRCTR                                                00109100
      COMMON /INOUT/NW,NR                                               00109200
      CHARACTER STRMCH*6                                                00109300
      DOUBLE PRECISION FLONRT                                           00109400
      INTEGER OUTPTS                                                    00109500
      DIMENSION STRMCH(2)                                               00109600
      DATA (STRMCH(I),I=1,2)/'INLET','OUTLET'/                          00109700
11    INPTS=0                                                           00109800
      OUTPTS=0                                                          00109900
      DO 10 I=1,NSTRMS                                                  00110000
         IF(NSCM(I,2) .EQ. 0) OUTPTS=OUTPTS+1                           00110100
         IF(NSCM(I,3) .EQ. 0) INPTS=INPTS+1                             00110200
10    CONTINUE                                                          00110300
      IF(OUTPTS .LE. 6 .AND. INPTS .LE. 6) GOTO 12                      00110400
      IF(OUTPTS .GT. 6) WRITE(NW,100) STRMCH(2)                         00110500
      IF(INPTS .GT. 6) WRITE(NW,100)  STRMCH(1)                         00110600
100   FORMAT(/'*** ERROR ***     TOTAL NUMBER OF ',A6,' STREAMS ',      00110700
     *        'EXCEEDS MAXIMUM VALUE OF 6.'/' PROCESS TOPOLOGY',        00110800
     *        'MUST NOW BE RESET.')                                     00110900
      CALL INISHL                                                       00111000
      CALL TOPOLI                                                       00111100
      CALL STNSCM                                                       00111200
      GOTO 11                                                           00111300
12    RETURN                                                            00111400
```

```
      END                                                                   00111500
      SUBROUTINE STNSCM                                                     00111600
C                                                                           00111700
C*************************************************************************  00111800
C                                                                           00111900
C   THE PURPOSE OF THIS ROUTINE IS TO SET THE STREAM CONNECTION MATRIX      00112000
C   GIVEN THE PROCESS MATRIX AND MAXUID.                                    00112100
C                                                                           00112200
C*************************************************************************  00112300
C                                                                           00112400
C   VARIABLES:                                                              00112500
C           NS    STREAM NUMBER                                             00112600
C       NSTRMS    NUMBER OF STREAMS IN THE PROCESS.                         00112700
C         NCOL    TRACKS COLUMN NUMBER IN THE PROCESS MATRIX.               00112800
C                                                                           00112900
C*************************************************************************  00113000
C                                                                           00113100
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)        00113200
     *           ,NMRCTR                                                    00113300
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)            00113400
     *            , ITRS,ITRT,MAXITR,CTOL                                 00113500
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                          00113600
     *            CMP(100)                                                00113700
      COMMON /INOUT/NW,NR                                                   00113800
15    NSTRMS =0                                                             00113900
      DO 10 I=1,MAXUID                                                      00114000
        NCOL=3                                                              00114100
11      IF(NPM(I,NCOL) .GT. 0) THEN                                         00114200
           NSCM(NPM(I,NCOL),1)=NPM(I,NCOL)                                  00114300
           NSCM(NPM(I,NCOL),3)=I                                            00114400
           NCOL=NCOL+1                                                      00114500
           GOTO 11                                                          00114600
        END IF                                                              00114700
        IF(NPM(I,NCOL) .LT. 0) THEN                                         00114800
           NSCM(IABS(NPM(I,NCOL)),1)=IABS(NPM(I,NCOL))                      00114900
           NSCM(IABS(NPM(I,NCOL)),2)=I                                      00115000
           NCOL=NCOL+1                                                      00115100
           IF(NCOL .EQ. 8) GOTO 10                                          00115200
           GOTO 11                                                          00115300
        END IF                                                              00115400
10    CONTINUE                                                              00115500
      DO 12 I=1,9                                                           00115600
        IF(NSCM(I,1) .NE. 0) NSTRMS=NSTRMS+1                                00115700
12    CONTINUE                                                              00115800
C                                                                           00115900
C   CHECK TO SEE THAT ONLY CONSECUTIVE NUMBERS (0-NSTRMS) ARE               00116000
C   ENTERED FOR STREAM NUMBERS.                                             00116100
C                                                                           00116200
      DO 13 I=1,MAXUID                                                      00116300
        DO 14 J=3,7                                                         00116400
          IF(IABS(NPM(I,J)) .GT. NSTRMS) THEN                               00116500
            WRITE(NW,100)                                                   00116600
            CALL INISHL                                                     00116700
            CALL TOPOLI                                                     00116800
            GOTO 15                                                         00116900
          END IF                                                            00117000
14      CONTINUE                                                            00117100
13    CONTINUE                                                              00117200
100   FORMAT(/' ****  ERROR  ****          STREAM NUMBERS NOT'              00117300
     *   ,' NUMBERED CONSECUTIVELY.'/' !!!!  REMEDY  !!!!      STREAM'      00117400
     *     ,' NUMBERS MUST BE NUMBERED CONSECUTIVELY FROM 1 TO TOTAL'       00117500
     *     ,' NUMBER OF STREAMS. PROCESS MUST BE RESET.')                   00117600
      RETURN                                                                00117700
      END                                                                   00117800
      SUBROUTINE EXEC                                                       00120100
C                                                                           00120200
C*************************************************************************  00120300
C                                                                           00120400
C   THE PURPOSE OF EXEC IS TO READ THE SECOND COLUMN OF THE PROCESS         00120500
C   MATRIX, THEN CALL THOSE UNIT OPERATIONS.                                00120600
C                                                                           00120700
C*************************************************************************  00120800
```

```
C                                                                       00120900
C   VARIABLES:                                                          00121000
C                                                                       00121100
C        NSTOP    TELLS EXEC WHEN TO RETURN TO THE MAIN PROG.          00121200
C                 NSTOP=0   KEEP CALCULATING. IF THE UNITS ARE         00121300
C                 EXAMINED AND ALL HAVE BEEN CALCULATED, NSTOP=1,      00121400
C                 THEN GOTO MAIN.                                      00121500
C                                                                       00121600
C***********************************************************************00121700
C                                                                       00121800
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00121900
     *        ,IPHASE,MAXCMP,INERT(5)                                   00122000
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00122100
     *          ,NMRCTR                                                 00122200
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00122300
     *             , ITRS,ITRT,MAXITR,CTOL                              00122400
      COMMON /INOUT/NW,NR                                               00122500
      DOUBLE PRECISION FLONRT                                           00122600
      INTEGER FLGERR,UNTFLG                                             00122700
      NSTOP=0                                                           00122800
      I=0                                                               00122900
15    I=I+1                                                             00123000
      IF(I .GT. MAXUID) THEN                                            00123100
         DO 18 II=1,5                                                   00123200
            IF(NFLGCV(II) .NE. 0) THEN                                  00123300
               I=0                                                      00123400
               GOTO 15                                                  00123500
            END IF                                                      00123600
18       CONTINUE                                                       00123700
      GOTO 99                                                           00123800
      END IF                                                            00123900
      IGOTO=NPM(I,2)                                                    00124000
      GOTO(10,11,12,13,14), IGOTO                                       00124100
C                                                                       00124200
C   I=UNIT NUMBER                                                       00124300
C                                                                       00124400
10    CALL MIXER(I)                                                     00124500
      GOTO 15                                                           00124600
11    CALL SPLTTR(I)                                                    00124700
      GOTO 15                                                           00124800
12    CALL PFR(I)                                                       00124900
      GOTO 15                                                           00125000
13    CALL CSTR(I)                                                      00125100
      GOTO 15                                                           00125200
14    CALL BATCH(I)                                                     00125300
      GOTO 15                                                           00125400
99    RETURN                                                            00125500
      END                                                               00125600
      SUBROUTINE SPPRMS(INDX,NOUT,NUMOUT)                               00125700
C                                                                       00125800
C***********************************************************************00125900
C                                                                       00126000
C   THE PURPOSE OF THIS ROUTINE IS TO ACCEPT STREAM FRACTION DATA       00126100
C   FROM THE USER.                                                      00126200
C   THIS ROUTINE MAY GROW TO ACCEPT ANY OTHER SPLITTER PARAMETERS.      00126300
C                                                                       00126500
C***********************************************************************00126600
C                                                                       00126700
C   VARIABLES:                                                          00126800
C                                                                       00126900
C      SPFRAC   HOLDS FRACTIONS FOR SPLIT STREAMS. BASED ON FRACTION    00127000
C               OF INLET STREAM THAT IS TO BE SPLIT TO AN OUTLET        00127100
C               STREAM. DIMENSIONED TO 9. THE POSITION IN SPFRAC        00127200
C               CORRESPONDS TO THE STREAM NUMBER.                       00127300
C                                                                       00127400
C***********************************************************************00127500
C                                                                       00127600
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00127700
     *          ,NMRCTR                                                 00127800
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00127900
     *             , ITRS,ITRT,MAXITR,CTOL                              00128000
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                      00128100
```

```
     *            CMP(100)                                                  00128200
      COMMON /INOUT/NW,NR                                                  00128300
      INTEGER FLGERR,UNTFLG                                                00128400
      DIMENSION NOUT(5)                                                    00128500
      SUM=0.                                                               00128600
      IF(NUMOUT .LE. 1) THEN                                               00129000
         WRITE(NW,101)                                                     00129100
101      FORMAT(' FATAL ERROR. PLEASE SPECIFY MORE THAN',                  00129200
     * ' ONE OUTLET STREAM FROM A SPLITTER')                               00129300
         STOP                                                              00129400
      END IF                                                               00129500
      NLESS1=NUMOUT-1                                                      00129600
      DO 10 I=1,NLESS1                                                     00129700
         WRITE(NW,100) NOUT(I)                                             00129800
100      FORMAT(' PLEASE SPECIFY FRACTION OF FEED STREAM TO PRODUCT'       00129900
     *          ,' STREAM # ',I2)                                          00130000
         READ(NR,*) SPFRAC(NOUT(I))                                        00130100
         CALL RERRLM(0.01,0.99,SPFRAC(NOUT(I)))                            00130200
         SUM=SUM+SPFRAC(NOUT(I))                                           00130300
10    CONTINUE                                                            00130400
      SPFRAC(NOUT(NUMOUT))=1.-SUM                                          00130500
      KX=NUMOUT                                                            00130600
C                                                                          00130700
C   NOW HAVE FRACTIONS STORED.                                             00130800
C                                                                          00130900
      RETURN                                                               00131000
      END                                                                  00131100
      SUBROUTINE STTEMP(INDX)                                              00131200
C                                                                          00131300
C***********************************************************************   00131400
C                                                                          00131500
C   THIS ROUTINE IS CALLED ONCE, AT THE BEGGINNG OF A UNIT COMPUTATION     00131600
C   TO REQUEST STREAM TEMPERATURES FOR THOSE STREAMS WHICH ARE PROCESS     00131700
C   FEED STREAMS.                                                          00131800
C                                                                          00131900
C***********************************************************************   00132000
C                                                                          00132100
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                   00132200
     *         ,IPHASE,MAXCMP,INERT(5)                                     00132300
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)       00132400
     *          ,NMRCTR                                                    00132500
      COMMON /INOUT/NW,NR                                                  00132600
      DOUBLE PRECISION FLONRT                                              00132700
      DO 10 I=1,NSTRMS                                                     00132800
         IF(NSCM(I,2) .EQ. 0 .AND. NSCM(I,3) .EQ. INDX) THEN               00132900
            IF(STRMID(I,10,1) .GT. 0.) GOTO 10                             00133000
            WRITE(NW,104) I                                                00133100
            READ (NR,*) STRMID(I,10,1)                                     00133200
            STRMID(I,10,1)=STRMID(I,10,1)+273.16                           00133300
         END IF                                                            00133400
 10   CONTINUE                                                            00133500
 104  FORMAT(' ENTER TEMPERATURE (C) FOR STREAM ',I2)                      00133600
      RETURN                                                               00133700
      END                                                                  00133800
      SUBROUTINE MIXER (INDX)                                              00133900
C                                                                          00134000
C***********************************************************************   00134100
C                                                                          00134200
C   THE PURPOSE OF THIS ROUTINE IS TO SIMULATE AN ADIABATIC MIXER.         00134300
C   A MAX OF FOUR INPUT STREAMS CAN BE MIXED TO MAKE ONE OUTLET STRM.      00134400
C                                                                          00134500
C***********************************************************************   00134600
C                                                                          00134700
C   VARIABLES:                                                             00134800
C                                                                          00134900
C         NIN  ARRAY THAT HOLDS THE STREAM NUMBERS OF ALL THE              00135000
C              INLET STRMS TO THE MIXER.                                   00135100
C        NOUT  ARRAY THAT HOLDS ALL THE OUTLET STREAM NUMBERS              00135200
C                                                                          00135300
C***********************************************************************   00135400
C                                                                          00135410
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                   00135500
```

```
     *         ,IPHASE,MAXCMP,INERT(5)                                        00135600
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)         00135700
     *            ,NMRCTR                                                    00135800
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)             00135900
     *             , ITRS,ITRT,MAXITR,CTOL                                   00136000
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                           00136100
     *            CMP(100)                                                   00136200
      COMMON /INOUT/NW,NR                                                    00136300
      DOUBLE PRECISION FLONRT                                                00136400
      INTEGER FLGERR,UNTFLG                                                  00136500
      DIMENSION NOUT(5),NIN(5)                                               00136600
      IF(UNTFLG(INDX,2) .EQ. 1) THEN                                         00137500
         RETURN                                                              00137700
      END IF                                                                 00137800
      CALL NOUTO(NOUT)                                                       00137900
      CALL NOUTO(NIN)                                                        00138000
C                                                                            00138100
C   FIND OUTLET STREAMS AND INLET STREAMS                                    00138200
C                                                                            00138300
      CALL FNDOUT(INDX,NOUT,NUMOUT)                                          00138400
      CALL FNDIN(INDX,NIN,NUMIN)                                             00138500
      CALL MIX(INDX,NIN,NUMIN,NOUT,NUMOUT)                                   00139300
      IF(NFLGCV(INDX) .EQ. 1) CALL STRMCV(INDX,NOUT,NUMOUT)                  00139400
      CALL TFACE(INDX,NIN,NUMIN)                                             00139500
C                                                                            00139700
C   SET UNTFLG =1 SO PROG KNOWS THAT UNIT HAS BEEN CALC'D AT LEAST           00139800
C   ONCE.                                                                    00139900
C                                                                            00140000
      UNTFLG(INDX,1)=1                                                       00140100
      CALL TFACE(INDX,NOUT,NUMOUT)                                           00141000
      RETURN                                                                 00142300
      END                                                                    00142400
      SUBROUTINE STCP                                                        00142500
C                                                                            00142510
C*************************************************************************** 00142600
C                                                                            00142700
C   THE PURPOSE OF THIS ROUTINE IS TO ACCEPT SPECIFIC HEAT VALUES FOR        00142800
C   1. LIQUID PHASE COMPONENTS  AND 2.(IF NEEDED) CMPS THAT ARE NOT          00142900
C   COVERED BY MY DATA.                                                      00143000
C                                                                            00143100
C   THE SPECIFIC HEATS(CAL/GMOL C) ARE STORED SEQUENTIALLY IN CPLIQ          00143200
C   FOLLOWING THE FIRST COLUMN IN NCOEF.                                     00143300
C                                                                            00143400
C*************************************************************************** 00143500
C                                                                            00143600
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                        00143700
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS       00143800
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)           00143900
     *         ,RXN(5,14)                                                    00144000
      COMMON /INOUT/NW,NR                                                    00144100
      COMMON /LIQDAT/CPLIQ(9)                                                00144200
      COMMON /LIBDAT/NLIB,NDUM,CPROPS(22,61)                                 00144300
      REAL*8 CPROPS                                                          00144400
      CHARACTER RXN                                                          00144500
      DO 10 I=1,9                                                            00144600
         IF(NCOEF(I,1) .EQ. 0) GOTO 11                                       00144700
         WRITE(NW,100) CPROPS(NCOEF(I,1),8),CPROPS(NCOEF(I,1),9)             00144800
100      FORMAT(' ENTER SPECIFIC HEAT (CAL/GMOL C) FOR COMPONENT ',2A4)      00144900
10    READ(NR,*) CPLIQ(I)                                                    00145000
11    RETURN                                                                 00145100
      END                                                                    00145200
      SUBROUTINE MIX(INDX,NIN,NUMIN,NOUT,NUMOUT)                             00145300
C                                                                            00145400
C*************************************************************************** 00145500
C                                                                            00145600
C   THIS ROUTINE TRANSFERS COMPONENT FLOW RATES FROM INLET STREAMS TO        00145700
C   THE CMP ARRAY. THE COMPONENT ID # IS USED AS THE INDX NUMBER IN CMP      00145800
C   I.E. IF AN ID # IS 12, CMP(12) WILL HOLD THE TOTAL INLET FLOW FOR        00145900
C   COMPONENT 12.                                                            00146000
C                                                                            00146100
C*************************************************************************** 00146200
C                                                                            00146300
```

```
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00146400
     *           ,NMRCTR                                               00146500
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00146600
     *         ,IPHASE,MAXCMP,INERT(5)                                 00146700
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                      00146800
     *            CMP(100)                                             00146900
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00147000
     *             , ITRS,ITRT,MAXITR,CTOL                             00147100
      INTEGER FLGERR,UNTFLG                                             00147200
      DOUBLE PRECISION FLONRT                                           00147300
      DIMENSION NIN(5),NOUT(5)                                          00147400
      DO 10 I=1,100                                                     00147600
10      CMP(I)=0.                                                       00147700
      I=1                                                               00147800
C                                                                       00147900
C   SEARCH INLET STREAMS FOR FLOWS > 0. PUT THESE INLET FLOWS INTO THE  00148000
C   CMP ARRAY.                                                          00148100
C                                                                       00148200
14    IF(NIN(I) .LE. 0) GOTO 13                                         00148300
         DO 11 J=1,9                                                    00148400
            IF(STRMID(NIN(I),J,2) .GT. 0.) CMP(INT(STRMID(NIN(I),J,1))) 00148500
     *      =CMP(INT(STRMID(NIN(I),J,1))) + STRMID(NIN(I),J,2)          00148600
11       CONTINUE                                                       00148700
         I=I + 1                                                        00148800
         GOTO 14                                                        00148900
13    CONTINUE                                                          00149000
      J=0                                                               00149200
      DO 12 I=1,100                                                     00149300
         IF(CMP(I) .GT. 0.) THEN                                        00149400
            J=J + 1                                                     00149500
            STRMID(NOUT(1),J,1)=FLOAT(I)                                00149600
            STRMID(NOUT(1),J,2)=CMP(I)                                  00149700
         END IF                                                         00149800
12    CONTINUE                                                          00149900
C                                                                       00150000
C   INSERT OUTLET STREAM TEMPERATURE IF ISOTHERMAL OPERATION            00150100
C                                                                       00150200
      IF(NOPCND(INDX) .EQ. 1) STRMID(NOUT(1),10,1)=STRMID(NIN(1),10,1)  00150300
      RETURN                                                            00150500
      END                                                               00150600
      SUBROUTINE STOPCD(INDX)                                           00150800
C                                                                       00150900
C*********************************************************************  00151000
C                                                                       00151100
C   THE PURPOSE OF THIS ROUTINE IS TO SET EACH UNIT'S OPERATING         00151200
C   CONDITION. IF ONE UNIT IS TO OPERATE NON-ISOTHERMALLY, THEN         00151300
C   ALL THE UNITS WILL BE TAGGED NOPCND=2 INITIALLY. THIS WAY           00151400
C   ALL STREAM ENTHALPIES WILL BE CALCULATED.                           00151500
C                                                                       00151600
C*********************************************************************  00151700
C                                                                       00151800
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00151900
     *           ,NMRCTR                                               00152000
      COMMON /INOUT/NW,NR                                               00152100
      CHARACTER UNTCHR*20                                               00152200
      DIMENSION UNTCHR(5)                                               00152300
      DATA (UNTCHR(I),I=1,5)/'MIXER','SPLITTER','PLUG FLOW RCTR',       00152400
     *                       'STIRRED TANK REACTOR','BATCH REACTOR'/    00152500
      WRITE(NW,100) INDX,UNTCHR(NPM(INDX,2))                            00152600
 100  FORMAT(/' SELECT UNIT #',I2,'-',A8,' OPERATING CONDITION:'/       00152700
     *   '      1) ISOTHERMAL'/                                         00152800
     *   '      2) NONISOTHERMAL'/)                                     00152900
      READ(NR,*) NOPCND(INDX)                                           00153000
      CALL IERRLM(1,2,NOPCND(INDX))                                     00153100
      IF(NOPCND(INDX) .EQ. 1) GOTO 10                                   00153200
      DO 11 I=1,MAXUID                                                  00153300
11       IF(NPM(I,2) .EQ. 2) NOPCND(I)=1                                00153400
10    RETURN                                                            00153500
      END                                                               00153600
      SUBROUTINE STOUTT(NIN,NOUT,NUMOUT)                                00153700
C                                                                       00153800
C*********************************************************************  00153900
```

```
C                                                                       00154000
C   THIS ROUTINE SETS THE TEMPERATURE FOR THE OUTLET STREAMS FROM A     00154100
C   SPLITTER. ONLY ONE STREAM IS ASSUMED TO FEED TO A SPLITTER.         00154200
C                                                                       00154300
C*********************************************************************  00154400
C                                                                       00154500
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00154600
     *         ,IPHASE,MAXCMP,INERT(5)                                  00154700
      DIMENSION NIN(5),NOUT(5)                                          00154800
      DOUBLE PRECISION FLONRT                                           00154900
      I=1                                                               00155000
      IF(STRMID(NIN(I),10,1) .GT. 0.) THEN      .                       00155100
         DO 12 J=1,NUMOUT                                               00155200
12          STRMID(NOUT(J),10,1)=STRMID(NIN(I),10,1)                    00155300
      END IF                                                            00155400
      RETURN                                                            00155500
      END                                                               00155600
      SUBROUTINE SPLTTR(INDX)                                           00155700
C                                                                       00155800
C*********************************************************************  00155900
C                                                                       00156000
C   THE PURPOSE OF THIS ROUTINE IS TO SIMULATE AN ADIABATIC STREAM      00156100
C   SPLITTER. ONE INPUT STREAM ONLY IS ALLOWED TO A SPLITTER. UP TO     00156200
C   FOUR OUTLET STREAMS ARE ALLOWED.                                    00156300
C                                                                       00156400
C*********************************************************************00156500
C                                                                       00156510
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00156600
     *         ,IPHASE,MAXCMP,INERT(5)                                  00156700
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00156800
     *           ,NMRCTR                                                00156900
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00157000
     *             , ITRS,ITRT,MAXITR,CTOL                            00157100
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                      00157200
     *            CMP(100)                                            00157300
      COMMON /INOUT/NW,NR                                               00157400
      DOUBLE PRECISION FLONRT                                           00157500
      INTEGER FLGERR,UNTFLG                                             00157600
      DIMENSION NOUT(5),NIN(5)                                          00157700
      IF(UNTFLG(INDX,2) .EQ. 1) THEN                                    00158800
         RETURN                                                         00159000
      END IF                                                            00159100
      CALL NOUTO(NOUT)                                                  00159400
      CALL NOUTO(NIN)                                                   00159500
      CALL FNDOUT(INDX,NOUT,NUMOUT)                                     00159600
      CALL FNDIN(INDX,NIN,NUMIN)                                        00159700
C                                                                       00159800
C   CALL STOUTT TO SET THE OUTLET TEMP IF ANY UNIT IS RUNNING NOPCND=2  00159900
C                                                                       00160000
      CALL STOUTT(NIN,NOUT,NUMOUT)                                      00160100
      CALL TFACE(INDX,NIN,NUMIN)                                        00160800
C                                                                       00160810
C   INDX=UNIT #                                                         00160900
C   NPM(INDX,3)=INLET STREAM TO SPLITTER                                00161000
C                                                                       00161100
      INLT=NPM(INDX,3)                                                  00161200
C                                                                       00161300
      CALL SPLT(INDX,NIN,NUMIN,NOUT,NUMOUT)                             00161400
C                                                                       00161500
C   CALCULATE OUTLET STREAM ENTHALPIES USING THE SPFRACS INSTEAD OF     00161600
C   CALCULATING THEM IN TFACE.                                          00161700
C                                                                       00161800
      DO 15 I=1,5                                                       00161900
         IF(NOUT(I) .EQ. 0) GOTO 16                                     00162000
         DO 17 K=1,9                                                    00162100
            IF(STRMID(INLT,K,5) .LT. 0.) GOTO 17                        00162200
            STRMID(NOUT(I),K,5)=STRMID(INLT,K,5)*SPFRAC(NOUT(I))        00162300
17       CONTINUE                                                       00162400
15    CONTINUE                                                          00162500
16    IF(NFLGCV(INDX) .EQ. 1) CALL STRMCV(INDX,NOUT,NUMOUT)             00162600
      CALL TFACE(INDX,NOUT,NUMOUT)                                      00162700
      UNTFLG(INDX,1)=1                                                  00163800
```

```
      RETURN                                                            00164000
      END                                                               00164100
      SUBROUTINE SPLT(INDX,NIN,NUMIN,NOUT,NUMOUT)                       00164200
C                                                                       00164300
C*********************************************************************  00164400
C                                                                       00164500
C   THIS ROUTINE IS USED IN  CONJUNCTION WITH THE SPLTTR MODULE. THIS   00164600
C   ROUTINE ASSUMES ONLY 1 INLET STREAM TO A SPLITTER.                  00164700
C                                                                       00164800
C*********************************************************************  00164900
C                                                                       00164910
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00165000
     *        ,IPHASE,MAXCMP,INERT(5)                                   00165100
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                      00165200
     *           CMP(100)                                               00165300
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00165400
     *             , ITRS,ITRT,MAXITR,CTOL                            00165500
      INTEGER FLGERR,UNTFLG                                             00165600
      DOUBLE PRECISION FLONRT                                           00165700
      DIMENSION NIN(5),NOUT(5)                                          00165800
      DO 10 I=1,100                                                     00165900
10       CMP(I)=0.                                                      00166000
      DO 11 J=1,9                                                       00166100
         IF(STRMID(NIN(1),J,2) .GT. 0.) THEN                            00166200
            CMP(INT(STRMID(NIN(1),J,1)))=STRMID(NIN(1),J,2)             00166300
            DO 12 K=1,NUMOUT                                            00166400
               STRMID(NOUT(K),J,1)=STRMID(NIN(1),J,1)                   00166500
12          STRMID(NOUT(K),J,2)=STRMID(NIN(1),J,2)*SPFRAC(NOUT(K))      00166600
         END IF                                                         00166700
11    CONTINUE                                                          00166800
      RETURN                                                            00166900
      END                                                               00167000
      SUBROUTINE NOUTO(NOUT)                                            00167100
C*********************************************************************  00167200
C                                                                       00167300
C   THE PURPOSE OF THIS PROGRAM IS TO INITIALIZE THE ARRAY NOUT(I)      00167400
C   TO 0. THIS PROGRAM IS CALLED AT THE BEGINNING OF EACH UNIT COMPU-   00167500
C   TATION.                                                             00167600
C                                                                       00167700
C*********************************************************************  00167800
     . DIMENSION NOUT(5)                                                00167900
      DO 10 I=1,5                                                       00168000
10       NOUT(I)=0                                                      00168100
      RETURN                                                            00168200
      END                                                               00168300
      SUBROUTINE FNDOUT(I,NOUT,NUMOUT)                                  00168400
C*********************************************************************  00168500
C                                                                       00168600
C   THE PURPOSE OF THIS ROUTINE IS TO FIND ALL OUTLET STREAMS FROM A    00168700
C   UNIT USING THE PROCESS MATRIX INFORMATION. THE OUTLET STREAM #'S     00168800
C   ARE STORED SEQUENTIALLY IN THE ARRAY NOUT(I). THIS ROUTINE IS       00168900
C   CALLED AT THE BEGINNING OF EACH UNIT OPERATION.                     00169000
C                                                                       00169100
C*********************************************************************  00169200
      COMMON/TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)     00169300
     *          ,NMRCTR                                                 00169400
      DIMENSION NOUT(5)                                                 00169500
      K=1                                                               00169600
      DO 10 J=3,7                                                       00169700
         IF(NPM(I,J) .LT. 0) THEN                                       00169800
            NOUT(K)=IABS(NPM(I,J))                                      00169900
            K=K+1                                                       00170000
         END IF                                                         00170100
10    CONTINUE                                                          00170200
      NUMOUT=K-1                                                        00170300
      RETURN                                                            00170400
      END                                                               00170500
      SUBROUTINE FNDREC(NIN,NUMIN)                                      00170600
C                                                                       00170700
C*********************************************************************  00170800
C                                                                       00170900
C    THE PURPOSE OF THIS ROUTINE IS TO CHECK TO SEE IF, AFTER DOING     00171000
```

```
C     A UNIT CALCULATION FOR THE FIRST TIME, THERE IS AN INPUT STREAM    00171100
C     THAT IS UNKNOWN I.E. TOTAL STREAM FLOW=-1. IF SO, THAT STREAM      00171200
C     IS TAGGED RECYCLE AND THE WHOLE PROCESS NEEDS TO GO THRU A         00171300
C     CONVERGE SUBROUTINE.                                               00171400
C                                                                        00171500
C***********************************************************************00171600
C                                                                        00171700
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                 00171800
     *         ,IPHASE,MAXCMP,INERT(5)                                   00171900
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)     00172000
     *           ,NMRCTR                                                 00172100
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)         00172200
     *            , ITRS,ITRT,MAXITR,CTOL                                00172300
      COMMON /INOUT/NW,NR                                                00172400
      DOUBLE PRECISION FLONRT                                            00172500
      INTEGER FLGERR,UNTFLG                                              00172600
      DIMENSION NIN(5)                                                   00172700
      DO 10 I=1,NUMIN                                                    00172800
         NS=NIN(I)                                                       00172900
         IF(STRMID(NS,1,2) .GT. 0.) GOTO 10                              00173200
         DO 11 J=1,MAXUID                                                00173300
11          NFLGCV(J)=1                                                  00173400
10    CONTINUE                                                           00173600
      RETURN                                                             00173800
      END                                                                00173900
      SUBROUTINE OUTPUT                                                  00174000
C                                                                        00174100
C***********************************************************************00174200
C                                                                        00174300
C   THIS ROUTINE PRINTS THE RESULTS OF THE PROCESS MODEL.                00174400
C                                                                        00174500
C***********************************************************************00174600
C                                                                        00174700
C   VARIABLES:                                                           00174800
C                                                                        00174900
C            KUNITS = 0 TO KEEP THE ORIGINAL UNITS                       00175000
C                   = 1 TO CONVERT TO UNITS OF GRAMS, GRAM-MOLES, KCAL   00175100
C                                                                        00175200
C***********************************************************************00175300
C                                                                        00175310
      COMMON /INOUT/NW,NR                                                00175400
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                 00175500
     *         ,IPHASE,MAXCMP,INERT(5)                                   00175600
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)     00175700
     *           ,NMRCTR                                                 00175800
      COMMON /CHARA/MODULE(5),CHRCHK(39),SEP(3),CHRUID(5)                00175900
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)         00176000
     *            , ITRS,ITRT,MAXITR,CTOL                                00176100
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                       00176200
     *             CMP(100)                                              00176300
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                    00176400
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS   00176500
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00176600
     *         ,RXN(5,14)                                                00176700
      COMMON /LIBDAT/NLIB,NDUM,CPROPS(22,61)                             00176800
      COMMON /ID/TITLE,NAME                                              00176900
      CHARACTER TITLE*60,NAME*20                                         00177000
      REAL*8 CPROPS                                                      00177100
      CHARACTER RXN                                                      00177200
      CHARACTER CHRCHK,SEP,MODULE*8,IDUM,CHRUID*2                        00177300
      DOUBLE PRECISION FLONRT                                            00177400
      INTEGER FLGERR,UNTFLG                                              00177500
      KUNITS=0                                                           00177600
      IF(STRMID(1,1,2) .LT. .01) KUNITS=1                                00177700
      WRITE(NW,105)                                                      00177800
105   FORMAT(//' PRINT RESULTS? (Y/N)'/)                                 00177900
      READ(NR,103) IDUM                                                  00178000
103   FORMAT(1X,A2)                                                      00178100
      IF(IDUM .EQ. ' '  .OR. IDUM .EQ. 'Y') NYESNO=1                     00178200
      IF(NYESNO .NE. 1 ) RETURN                                          00178300
      WRITE(NW,108) NAME,TITLE                                           00178400
108   FORMAT(///'   NAME: ',A20,/'   TITLE: ',A60/)                      00178500
```

```
      DO 10 I=1,9                                                       00178600
         IF(STRMID(I,1,1) .GT. 0.) THEN                                 00178700
            T=STRMID(I,10,1)-273.15                                     00178800
            WRITE(NW,100) I,T                                           00178900
100         FORMAT(//'*****************   STREAM NUMBER ',I1,' ******', 00179000
     *              '***********'/' TEMPERATURE (C) :',F12.4)           00179100
            IF(KUNITS .EQ. 0) WRITE(NW,102)                             00179200
102    FORMAT(/'  CMP  ',6X,'MOL FLOW',8X,'MASS FLOW',11X,'X',14X,'H',   00179300
     *      11X,'W'/11X,'KG-MOLS/MIN',8X,' KG/MIN',26X,'MCAL'/)         00179400
            IF(KUNITS .EQ. 1) WRITE(NW,120)                             00179500
120    FORMAT(/'  CMP  ',6X,'MOL FLOW',8X,'MASS FLOW',11X,'X',14X,'H',   00179600
     *      11X,'W'/11X,' G-MOLS/MIN',8X,'  G/MIN',26X,'KCAL'/)         00179700
            J=1                                                         00179800
11          IF(STRMID(I,J,1) .GT. 0 .AND. J .LT. 10) THEN               00179900
             L=INT(STRMID(I,J,1))                                       00180000
             IF(KUNITS .EQ. 1) THEN                                     00180100
                STRMID(I,J,2)=STRMID(I,J,2)*1000.                       00180200
                STRMID(I,J,3)=STRMID(I,J,3)*1000.                       00180300
                STRMID(I,J,4)=STRMID(I,J,4)*1000.                       00180400
             END IF                                                     00180500
             WRITE(NW,101) CPROPS(8,L),CPROPS(9,L),(STRMID(I,J,K),K=2,6)00180600
101          FORMAT(1X,2(A4),3X,F9.4,5X,F12.4,7X,F7.4,2X,F14.4,5X,F7.4) 00180700
             J=J+1                                                      00180800
             IF(J .EQ. 11) GOTO 10                                      00180900
             GOTO 11                                                    00181000
            END IF                                                      00181100
         END IF                                                         00181200
10     CONTINUE                                                         00181300
       IF(NMRCTR .GT. 0) THEN                                           00181400
          DO 12 I=1,5                                                   00181500
             IF(NPM(I,2) .GT. 2) THEN                                   00181600
                WRITE(NW,104) I,CHRUID(NPM(I,2)),RCTVOL(I),X(I)         00181700
             END IF                                                     00181800
12        CONTINUE                                                      00181900
       END IF                                                           00182000
104    FORMAT(//' UNIT #',I2,'   UNIT: ',A2,' REACTOR VOLUME (L) : ',   00182100
     * F14.3,/'                                CONVERSION : ',F14.3)     00182200
       I=1                                                              00182300
       WRITE(NW,106) I,(RXN(1,J),J=1,14)                                00182400
106    FORMAT(/' REACTIONS:   ',I1,')',14A1)                            00182500
       IF(NRXNS .GT. 1) THEN                                            00182600
          DO 13 I=2,NRXNS                                               00182700
13           WRITE(NW,107) I,(RXN(I,J),J=1,14)                          00182800
       END IF                                                           00182900
107    FORMAT( '              ',I1,')',14A1)                            00183000
       WRITE(NW,111)                                                    00183010
111    FORMAT(/' DO YOU WISH TO VIEW YIELD DATA?  (Y/N)')               00183020
       READ(NR,121) IDUM                                                00183030
121    FORMAT(A1)                                                       00183031
       IF(IDUM .EQ. 'Y' .OR. IDUM .EQ. ' ') THEN                        00183040
       WRITE(NW,109)                                                    00183050
109    FORMAT(' YIELD DATA WILL BE PRINTED FOR TWO BASIS''. FRACTIONAL', 00183060
     *' YIELD (FY)'/' IS DEFINED AS: MOLES OF PRODUCT PRODUCED/MOLE ',   00183070
     *  'KEY REACTANT REACTED.'/' YIELD BASED ON FEED (YBOF) IS DE',     00183080
     *  'FINED AS: MOLES PRODUCT PRODUCED/MOLE KEY REACTANT REACTED.')   00183090
       CALL YIELD                                                       00183091
       END IF                                                           00183092
14     WRITE(NW,110)                                                    00183093
110    FORMAT(/' DO YOU WISH TO VIEW DATA GENERATED DURING INTEGRATION?' 00183094
     *,' (Y/N)')                                                        00183095
       READ(NR,121) IDUM                                                00183096
       IF(IDUM .EQ. 'Y' .OR. IDUM .EQ. ' ') CALL INTDTA                 00183097
       RETURN                                                           00183279
       END                                                              00183280
       SUBROUTINE CALCHT(STNUM,N)                                       00183489
C                                                                       00183490
C***********************************************************************00183500
C                                                                       00183600
C   THIS ROUTINE CALCULATES TOTAL STREAM ENTHALPIES GIVEN COMPONENT     00183700
C   ENTHALPIES.                                                         00183800
C                                                                       00184100
C***********************************************************************00184200
```

```
C                                                                       00184300
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00184400
     *        ,IPHASE,MAXCMP,INERT(5)                                   00184500
      DOUBLE PRECISION FLONRT                                           00184600
      INTEGER STNUM(5)                                                  00184700
      DO 10 I=1,N                                                       00184800
         NS=STNUM(I)                                                    00184900
         SUMH=0.                                                        00185100
         DO 11 J=1,9                                                    00185200
            IF(INT(STRMID(NS,J,5)) .LT. 0) GOTO 12                      00185300
            SUMH=SUMH+STRMID(NS,J,5)                                    00185400
11       CONTINUE                                                       00185500
12       STRMID(NS,10,7)=SUMH                                           00185600
10    CONTINUE                                                          00185800
      RETURN                                                            00185900
      END                                                               00186000
      SUBROUTINE CALCMT(STNUM,N)                                        00186100
C                                                                       00186200
C***********************************************************************00186300
C                                                                       00186400
C   THIS ROUTINE CALCULATES TOTAL STREAM MASS AND MOLAR FLOWS GIVEN     00186500
C   COMPONENT FLOWS. IT ALSO CALCULATES COMPONENT MASS AND MOLE         00186600
C   FRACTIONS.                                                          00186700
C                                                                       00186800
C***********************************************************************00186900
C                                                                       00187000
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00187100
     *        ,IPHASE,MAXCMP,INERT(5)                                   00187200
      DOUBLE PRECISION FLONRT                                           00187300
      INTEGER STNUM(5)                                                  00187400
      DO 10 I=1,N                                                       00187500
         NS=STNUM(I)                                                    00187600
         SUMMAS=0.                                                      00187700
         SUMMOL=0.                                                      00187800
         DO 11 J=1,9                                                    00187900
            IF(STRMID(NS,J,1) .LT. 0.) GOTO 12                          00188000
            SUMMAS=SUMMAS+STRMID(NS,J,3)                                00188100
11       SUMMOL=SUMMOL+STRMID(NS,J,2)                                   00188200
12       STRMID(NS,10,4)=SUMMAS                                         00188300
         STRMID(NS,10,3)=SUMMOL                                         00188400
         DO 13 K=1,9                                                    00188700
            IF(STRMID(NS,K,2) .LE. 0.) GOTO 10                          00188800
            STRMID(NS,K,6)=STRMID(NS,K,3)/SUMMAS                        00188900
13          STRMID(NS,K,4)=STRMID(NS,K,2)/SUMMOL                        00189000
10    CONTINUE                                                          00189100
      RETURN                                                            00189200
      END                                                               00189300
      SUBROUTINE NOCONV(TOL,N)                                          00189400
C                                                                       00189500
C***********************************************************************00189600
C                                                                       00189700
C   THE PURPOSE OF THIS ROUTINE IS TO ALERT THE USER THAT AN ITERATIVE  00189800
C   CALCULATION HAS EXCEEDED THE MAXIMUM NUMBER OF ITERATIONS.          00189900
C                                                                       00190000
C***********************************************************************00190100
C                                                                       00190200
C   THE ARGUMENTS ARE:                                                  00190300
C                                                                       00190400
C          TOL = LAST VALUE OF THE TOLERANCE. WILL SHOW HOW CLOSE       00190500
C                THE CALCULATIONS CAME TO THE FINAL RESULT.             00190600
C            N = NUMBER IS USED TO TELL NOCONV WHERE THE                00190700
C                CONVERGENCE PROBLEM IS.                                00190800
C                                                                       00190900
C***********************************************************************00190910
C                                                                       00190920
      COMMON /INOUT/NW,NR                                               00191000
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00191100
     *             , ITRS,ITRT,MAXITR,CTOL                              00191200
      INTEGER FLGERR,UNTFLG                                             00191300
      CHARACTER DUM                                                     00191400
      WRITE(NW,100) MAXITR,TOL                                          00191500
100   FORMAT('******** MAXIMUM NUMBER OF ITERATIONS (',I2,') EXCEEDED.' 00191600
```

```
     * /' ******** THE LAST VALUE OF TOL WAS',F8.5/)                    00191700
      GOTO (1,2,3,4,5,6),N                                              00191800
1     WRITE(NW,104)                                                     00191900
104   FORMAT(' LAST CALCULATION OCCURRED IN ROUTINE MIXER')             00192000
      GOTO 7                                                            00192100
2     WRITE(NW,105)                                                     00192200
105   FORMAT(' LAST CALCULATION OCCURRED IN ROUTINE SPLTTR')            00192300
      GOTO 7                                                            00192400
3     WRITE(NW,106)                                                     00192500
106   FORMAT(' LAST CALCULATION OCCURRED IN ROUTINE PFR')               00192600
      GOTO 7                                                            00192700
4     WRITE(NW,107)                                                     00192800
107   FORMAT(' LAST CALCULATION OCCURRED IN ROUTINE CSTR')              00192900
      GOTO 7                                                            00193000
5     WRITE(NW,108)                                                     00193100
108   FORMAT(' LAST CALCULATION OCCURRED IN ROUTINE BATCH')             00193200
      GOTO 7                                                            00193300
6     WRITE(NW,109)                                                     00193400
109   FORMAT(' LAST CALCULATION OCCURRED IN ROUTINE FNDTMP')            00193500
7     WRITE(NW,101)                                                     00193600
101   FORMAT(' DO YOU WANT THE RESULTS AND THE LAST VALUE OF THE ',     00193700
     *       'CALCULATED TOLERANCE PRINTED?  (Y/N)')                    00193800
      READ(NR,102) DUM                                                  00193900
102   FORMAT(A2)                                                        00194000
      IF(DUM .EQ. 'Y') CALL OUTPUT                                      00194100
      IF(DUM .EQ. 'Y') WRITE(NW,103) TOL                                00194200
103   FORMAT(' FINAL TOLERANCE IS: ',F10.5)                             00194300
      RETURN                                                            00194400
      END                                                               00194500
      SUBROUTINE FNDTMP(INDX,STNUM)                                     00194600
C                                                                       00194700
C*********************************************************************  00194800
C                                                                       00194900
C   THE PURPOSE OF THIS ROUTINE IS TO FIND THE OUTLET TEMPERATURE FOR   00195000
C   A STREAM FROM A MIXER GIVEN THE INLET CONDITIONS. ONCE THE CORRECT  00195100
C   OUTLET TEMPERATURE IS FOUND, COMPONENT AND STREAM ENTHALPIES AND    00195200
C   THE OUTLET TEMPERATURE ARE PLACED IN THE STRMID ARRAY.              00195300
C                                                                       00195400
C*********************************************************************  00195500
C                                                                       00195600
C   VARIABLES:                                                          00195700
C                                                                       00195800
C       STNUM  HOLDS THE STREAM NUMBERS(IN OR OUT)                      00195900
C                                                                       00196000
C        ITRT = ITERATION NUMBER FOR THE IMPLICIT CALCULATION OF        00196100
C               TEMPERATURE (K) FROM AN ENTHALPY BALANCE                00196200
C        ITRS = ITERATION NUMBER FOR STREAM CONVERGENCE                 00196300
C                                                                       00196400
C      MAXITR = MAXIMUM ALLOWABLE NUMBER OF ITERATIONS. THIS NUMBER     00196500
C               IS SET IN THE TOPOLI ROUTINE.                           00196600
C                                                                       00196700
C*********************************************************************  00196710
C                                                                       00196720
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00196800
     *        ,IPHASE,MAXCMP,INERT(5)                                   00196900
      COMMON /LIBDAT/NLIB,NDUM,CPROPS(22,61)                            00197000
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00197100
     *             , ITRS,ITRT,MAXITR,CTOL                              00197200
      DOUBLE PRECISION FLONRT                                           00197300
      INTEGER FLGERR,UNTFLG,STNUM(5)                                    00197400
      REAL*8 CPROPS                                                     00197500
      DIMENSION NIN(5)                                                  00197600
      DATA TO/0./                                                       00197700
C                                                                       00197800
C   FIND THE INLET STREAMS TO THE UNIT                                  00197900
C                                                                       00198000
      CALL NOUTO(NIN)                                                   00198100
      CALL FNDIN(INDX,NIN,NUMIN)                                        00198200
      TXM=0.                                                            00198300
      TOTLH=0.                                                          00199200
      TOTLM=0.                                                          00199300
      DO 10 I=1,NUMIN                                                   00199400
```

```
         IF(STRMID(NIN(I),1,2) .LT. 0.) GOTO 10                          00199500
         TXM=TXM+STRMID(NIN(I),10,3)*STRMID(NIN(I),10,1)                 00199600
         TOTLM=TOTLM+STRMID(NIN(I),10,3)                                 00199700
         TOTLH=TOTLH+STRMID(NIN(I),10,7)                                 00199800
10     CONTINUE                                                          00199900
C                                                                        00200200
C    GUESS T                                                             00200300
C                                                                        00200400
       T=TXM/TOTLM                                                       00200500
       ITR=0                                                             00200700
16     HOUT=0.                                                           00200900
       HDER=0.                                                           00201000
       ITR=ITR+1                                                         00201100
C                                                                        00201200
C    IF ITERATIONS EXCEED MAXITR CALL NOCONV (NO CONVERGENCE)            00201300
C                                                                        00201400
       IF(ITR .GT. MAXITR) THEN                                          00201600
          CALL NOCONV(TOL,6)                                             00201700
          STOP                                                           00201800
       END IF                                                            00202000
       I=1                                                               00202100
15     IF(STNUM(I) .NE. 0) THEN                                          00202200
          J=1                                                            00202300
14        NCMP=INT(STRMID(STNUM(I),J,1))                                 00202400
          SPCH=0.                                                        00202500
          SPHDER=0.                                                      00202600
          DO 12 II=1,7                                                   00202900
             SPHDER=SPHDER+CPROPS(II+15,NCMP)*T**(II-1)                  00203000
12        SPCH=SPCH+CPROPS(II+15,NCMP)*(T**(II)-TO**(II))/II             00203100
          H=SPCH*STRMID(STNUM(I),J,2)/1000.                              00203200
          HD=STRMID(STNUM(I),J,2)*SPHDER/1000.                           00203300
          STRMID(STNUM(I),J,5)=H                                         00203400
          HOUT=HOUT+H                                                    00203500
          HDER=HDER+HD                                                   00203600
          J=J+1                                                          00203700
          IF(J .EQ. 10) GOTO 13                                          00203800
          IF(STRMID(STNUM(I),J,2) .LE. 0.) GOTO 13                       00203900
          GOTO 14                                                        00204000
13        I=I+1                                                          00204100
          GOTO 15                                                        00204200
       END IF                                                            00204300
       TOL=(TOTLH-HOUT)/TOTLH                                            00204400
       IF(ABS(TOL) .LE. CTOL) GOTO 20                                    00204600
       T=T-(HOUT-TOTLH)/HDER                                             00204700
       GOTO 16                                                           00204800
20      DO 18 I=1,5                                                      00204900
          IF(STNUM(I) .EQ. 0.) GOTO 19                                   00205000
18        STRMID(STNUM(I),10,1)=T                                        00205100
19     RETURN                                                            00205300
       END                                                               00205400
       SUBROUTINE TFACE(INDX,STNUM,N)                                    00205500
C                                                                        00205600
C************************************************************************ 00205700
C                                                                        00205800
C    THIS ROUTINE IS RESPONSIBLE FOR CALCULATING THERMO PROPERTIES OF    00205900
C    COMPONENTS AND INTERFACING WITH THE THERMO DATA BANK(S).            00206000
C    TFACE ASSUMES THAT THE COMPONENT ID NUMBERS AND COMPONENT MOLAR     00206100
C    FLOW RATES ARE IN THE STRMID ARRAY.                                 00206200
C                                                                        00206300
C************************************************************************ 00206400
C                                                                        00206500
C    VARIABLES:                                                          00206600
C                                                                        00206700
C        INDX  =UNIT #                                                   00206800
C       STNUM  ARRAY THAT HOLDS THE STREAM NUMBERS (IN SEQUENTIAL ORDER) 00206900
C              THAT TFACE IS TO WORK ON.                                 00207000
C           N  THE NUMBER OF STREAMS THAT TFACE IS WORKING ON.           00207100
C                                                                        00207200
C************************************************************************ 00207210
C                                                                        00207220
       COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00207300
      *         ,IPHASE,MAXCMP,INERT(5)                                  00207400
```

```
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)     00207500
     *            ,NMRCTR                                               00207600
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                       00207700
     *             CMP(100)                                              00207800
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)         00207900
     *             , ITRS,ITRT,MAXITR,CTOL                               00208000
      COMMON /LIBDAT/NLIB,NDUM,CPROPS(22,61)                             00208100
      COMMON /INOUT/NW,NR                                                00208200
      DOUBLE PRECISION FLONRT                                            00208300
      REAL*8 CPROPS                                                      00208400
      INTEGER FLGERR,UNTFLG,STNUM(5)                                     00208500
      DATA TO/0./                                                        00208600
C                                                                        00208700
C   CALCULATE MASS,MASS FRACTION, AND MOLE FRACTION                      00208800
C                                                                        00208900
      DO 20 I=1,N                                                        00209300
         NS=STNUM(I)                                                     00209400
         J=1                                                             00209600
14       IF(STRMID(NS,J,2) .GT. 0.) THEN                                 00209700
            NC=INT(STRMID(NS,J,1))                                       00209800
            STRMID(NS,J,3)=STRMID(NS,J,2)*CPROPS(14,NC)                  00210000
            J=J+1                                                        00210100
            IF(J .EQ. 10) GOTO 20                                        00210200
            GOTO 14                                                      00210300
         END IF                                                          00210400
20    CONTINUE                                                           00210500
      CALL CALCMT(STNUM,N)                                               00210600
      IF(NPM(INDX,2) .EQ. 1) THEN                                        00210700
         IF(N .GT. 1) GOTO 9                                             00210800
         IF(NOPCND(INDX) .EQ. 2) CALL FNDTMP(INDX,STNUM)                 00210900
      END IF                                                             00211000
9     DO 10 I=1,N                                                        00211100
         NS=STNUM(I)                                                     00211200
         IF(NS .EQ. 0) GOTO 13                                           00211300
         IF(STRMID(NS,10,1) .LE. 0.) GOTO 10                             00211400
         T=STRMID(NS,10,1)                                               00211500
         DO 11 J=1,9                                                     00211600
            IF(STRMID(NS,J,2) .LE. 0.) GOTO 10                           00211700
            NC=INT(STRMID(NS,J,1))                                       00211800
C                                                                        00211900
C   CALCULATE SPECIFIC ENTHALPY (CAL/G-MOLE)                             00212000
C                                                                        00212100
            SPCH=0.                                                      00212200
            DO 12 II=1,7                                                 00212300
12             SPCH=SPCH+CPROPS(II+15,NC)*(T**(II)-TO**(II))/II          00212400
C                                                                        00212500
C   CALCULATE COMPONENT ENTHALPY (MCAL)                                  00212600
C                                                                        00212700
            STRMID(NS,J,5)=SPCH*STRMID(NS,J,2)/1000.                     00212800
11       CONTINUE                                                        00212900
10    CONTINUE                                                           00213000
13    CALL CALCHT(STNUM,N)                                               00213100
      RETURN                                                             00214100
      END                                                                00214200
C_____________________________________________________________________ 00214400
C                                                                        00214500
C***************    BLOCK DATA SUBPROGRAMS    ************************* 00214600
C                                                                        00214700
C------------------------------------------------------------------------ 00214900
      BLOCK DATA                                                         00215100
      COMMON /CHARA/MODULE,CHRCHK,SEP,CHRUID                             00215200
      CHARACTER MODULE*8,CHRCHK,SEP,CHRUID*2                             00215201
      DIMENSION MODULE(5),CHRCHK(39),SEP(3),CHRUID(5)                    00215210
      DATA (MODULE(I),I=1,5)/'MIXER','SPLITTER','PFR','CSTR','BATCH'/    00215300
      DATA (CHRUID(I),I=1,5)/'MX','SP','PF','CS','BR'/                   00215400
      DATA (SEP(I),I=1,3)/'+','=','>'/                                   00215500
      DATA (CHRCHK(I),I=1,39)/'A','B','C','D','E','F','G','H','I','J',  00215600
     *     'K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y', 00215700
     *     'Z','1','2','3','4','5','6','7','8','9','+','>','=',' '/     00215800
      END                                                                00215900
      BLOCK DATA                                                         00216000
C------------------------------------------------------------------------C00216100
```

```
C                                                                       00216200
C      PURPOSE --- THIS ROUTINES LOADS VARIOUS COMMON BLOCKS WITH       00216300
C                  DATA WHICH DOES NOT CHANGE DURING THE CALCULATIONS   00216400
C                                                                       00216500
C            THE ORDER OF THE LIBRARY DATA IS AS FOLLOWS:               00216600
C                                                                       00216700
C            XPRPXX(K)                 QUANTITY                       00216800
C            ---------              -------------                     00216900
C               1-7                 COMPONENT NAME                     00217000
C               8-9                 SYNONYM                            00217100
C                10                 PC (BAR)                           00217200
C                11                 VC (CM**3/MOL)                     00217300
C                12                 TC (KELVIN)                        00217400
C                13                 ACENTRIC FACTOR                    00217500
C                14                 MOLECULAR WEIGHT                   00217600
C                15                 T(NBP) (KELVIN)                    00217700
C              16-22                IDEAL HEAT CAPACITY COEFFICIENTS   00217800
C                                   WHERE CPO=SUM(C(J)*T**(J-1)) J=1,7 00217900
C                                   WHERE T HAS UNITS OF (K) AND CPO   00218000
C                                   HAS UNITS OF CAL/G-MOLE (C)        00218100
C                                                                       00218200
C      CODED BY:   J. F. ELY  &   MODIFIED(1)  BY: K.A.M. GASEM         00218300
C                             &   MODIFIED(2)  BY: W.J. VEDDER          00218400
C                                                                       00218500
C                  NATIONAL BUREAU OF STANDARDS                         00218600
C                  NATIONAL ENGINEERING LABORATORY                      00218700
C                  THERMOPHYSICAL PROPERTIES DIVISION                   00218800
C                  BOULDER, COLORADO 80303                              00218900
C                                                                       00219000
C      VERSION GPA2.1 - - 8/12/80                                       00219100
C      VERSION MODIFIED(1)- -   12/83                                   00219200
C      VERSION MODIFIED(2)- -   10/85                                   00219300
C                                                                       00219400
C-----------------------------------------------------------------------C00219500
C                                                                       00219600
      IMPLICIT REAL*8 (X)                                               00219700
C     DOUBLE PRECISION X ,XN                                            00219800
C                                                                       00219900
      COMMON /LIBDAT/    NLIB, NDUM, XPRPO1(22), XPRPO2(22), XPRPO3(22),00220000
     1       XPRPO4(22), XPRPO5(22), XPRPO6(22), XPRPO7(22), XPRPO8(22),00220100
     1       XPRPO9(22), XPRP10(22), XPRP11(22), XPRP12(22), XPRP13(22),00220200
     2       XPRP14(22), XPRP15(22), XPRP16(22), XPRP17(22), XPRP18(22),00220300
     3       XPRP19(22), XPRP20(22), XPRP21(22), XPRP22(22), XPRP23(22),00220400
     4       XPRP24(22), XPRP25(22), XPRP26(22), XPRP27(22), XPRP28(22),00220500
     5       XPRP29(22), XPRP30(22), XPRP31(22), XPRP32(22), XPRP33(22),00220600
     6       XPRP34(22), XPRP35(22), XPRP36(22), XPRP37(22), XPRP38(22),00220700
     7       XPRP39(22), XPRP40(22), XPRP41(22), XPRP42(22), XPRP43(22),00220800
     8       XPRP44(22), XPRP45(22), XPRP46(22), XPRP47(22), XPRP48(22),00220900
     9       XPRP49(22), XPRP50(22), XPRP51(22), XPRP52(22), XPRP53(22),00221000
     A       XPRP54(22), XPRP55(22), XPRP56(22), XPRP57(22), XPRP58(22),00221100
     B       XPRP59(22), XPRP60(22), XPRP61(22)                         00221200
C                                                                       00221300
C_____________________   ---------------------   _____________________  00221400
C                        LIBRARY COMPONENT DATA                         00221500
C-----------------------  _____________________  -----------------------00221600
      DATA NLIB / 61 /                                                  00221700
C                                                                       00221800
      DATA     XPRPO1           / 'HYDR', 'OGEN', '    ', '    ', '    ',00221900
     1   '    ', '    ', 'H2  ', '    ',       12.760D0,       64.000D0,00222000
     2       32.980D0,       0.00000D0,        2.020D0,       20.280D0,00222100
     3    .67501459D+01,  .38754626D-03,  .31844129D-17, -.87429101D-24,00222200
     4    .18934493D-12, -.95360879D-16,  .10839431D-19/                00222300
C                                                                       00222400
      DATA     XPRPO2           / 'METH', 'ANE ', '    ', '    ', '    ',00222500
     1   '    ', '    ', 'CH4 ', '    ',       45.387D0,       97.752D0,00222600
     2       190.555D0,       .011314D0,       16.043D0,       111.630D0,00222700
     3    .76692963D+01,  .33677582D-14,  .97978154D-05,  .16480983D-07,00222800
     4   -.19759199D-10, -.12233376D-21,  .31782271D-17/                00222900
C                                                                       00223000
      DATA     XPRPO3           / 'ETHA', 'NE  ', '    ', '    ', '    ',00223100
     1   '    ', '    ', 'C2H6', '    ',       48.077D0,       147.060D0,00223200
     2       305.330D0,       .10038D0,        30.070D0,       184.550D0,00223300
```

```
     3    .77063866D+01,  .44587991D-05,  .80959840D-04, -.85756975D-07,00223400
     4    .13183583D-10,  .21689626D-13, -.85251740D-17/                00223500
C                                                                        00223600
      DATA    XPRPO4          / 'PROP', 'ANE ', '    ', '    ', '    ',00223700
     1  '    ', '    ', 'C3H8', '    ',       41.914D0,      201.610D0,00223800
     2       369.820D0,       .15418D0,       44.097D0,      231.050D0,00223900
     3    .54902525D+01,  .40763758D-01,  .95793455D-05, -.92506902D-08,00224000
     4   -.71806497D-11,  .42920084D-18,  .26266881D-17/                00224100
C                                                                        00224200
      DATA    XPRPO5          / 'ISOB', 'UTAN', 'E   ', '    ', '    ',00224300
     1  '    ', '    ', 'I-C4', 'H10 ',       36.003D0,      263.000D0,00224400
     2       408.150D0,       .18300D0,       58.124D0,      261.420D0,00224500
     3    .83448658D+01,  .40337532D-02,  .28869652D-03, -.60188092D-06,00224600
     4    .57571458D-09, -.27078809D-12,  .50258511D-16/                00224700
C                                                                        00224800
      DATA    XPRPO6          / 'N-BU', 'TANE', '    ', '    ', '    ',00224900
     1  '    ', '    ', 'N-C4', 'H10 ',       37.465D0,      256.410D0,00225000
     2       425.160D0,       .20038D0,       58.124D0,      272.650D0,00225100
     3    .86549559D+01,  .34204803D-01,  .82496176D-04, -.97523439D-07,00225200
     4   -.35770071D-11,  .46492802D-13, -.16554141D-16/                00225300
C                                                                        00225400
      DATA    XPRPO7          / 'ISOP', 'ENTA', 'NE  ', '    ', '    ',00225500
     1  '    ', '    ', 'I-C5', 'H12 ',       33.368D0,      306.000D0,00225600
     2       460.430D0,       .22700D0,       72.151D0,      301.030D0,00225700
     3    .61029072D+01,  .78716077D-01,  .21259122D-06, -.20717458D-07,00225800
     4    .17894175D-10, -.30123834D-13,  .15028683D-16/                00225900
C                                                                        00226000
      DATA    XPRPO8          / 'N-PE', 'NTAN', 'E   ', '    ', '    ',00226100
     1  '    ', '    ', 'N-C5', 'H12 ',       33.319D0,      313.600D0,00226200
     2       469.750D0,       .25109D0,       72.151D0,      309.210D0,00226300
     3    .10821614D+02,  .30187967D-01,  .17717678D-03, -.30306205D-06,00226400
     4    .20500044D-09, -.57300484D-13,  .37303568D-17/                00226500
C                                                                        00226600
      DATA    XPRPO9          / 'NEO-', 'PENT', 'ANE ', '    ', '    ',00226700
     1  '    ', '    ', 'NEO-', 'C5  ',       31.572D0,      303.000D0,00226800
     2       433.780D0,       .19700D0,       72.151D0,      282.630D0,00226900
     3    .51988869D+01,  .68223744D-01,  .77955679D-04, -.12658923D-06,00227000
     4    .49657490D-11,  .61782974D-13, -.23947987D-16/                00227100
C                                                                        00227200
      DATA    XPRP10          / 'N-HE', 'XANE', '    ', '    ', '    ',00227300
     1  '    ', '    ', 'N-C6', 'H14 ',       29.884D0,      373.220D0,00227400
     2       507.889D0,       .29777D0,       86.178D0,      341.880D0,00227500
     3    .10736446D+02,  .64479391D-01,  .87557824D-04, -.11685574D-06,00227600
     4   -.12019797D-11,  .53278033D-13, -.19076817D-16/                00227700
C                                                                        00227800
      DATA    XPRP11          / 'N-HE', 'PTAN', 'E   ', '    ', '    ',00227900
     1  '    ', '    ', 'N-C7', 'H16 ',       26.997D0,      431.970D0,00228000
     2       540.140D0,       .34991D0,      100.206D0,      371.580D0,00228100
     3    .15205618D+02,  .22270435D-01,  .36769377D-03, -.72308061D-06,00228200
     4    .64010192D-09, -.27923771D-12,  .48332058D-16/                00228300
C                                                                        00228400
      DATA    XPRP12          / 'N-OC', 'TANE', '    ', '    ', '    ',00228500
     1  '    ', '    ', 'N-C8', 'H18 ',       24.650D0,      490.000D0,00228600
     2       568.820D0,       .39953D0,      114.232D0,      398.820D0,00228700
     3    .13014803D+02,  .92904278D-01,  .95691535D-04, -.13010358D-06,00228800
     4   -.18629703D-10,  .77087455D-13, -.26357422D-16/                00228900
C                                                                        00229000
      DATA    XPRP13          / 'N-NO', 'NANE', '    ', '    ', '    ',00229100
     1  '    ', '    ', 'N-C9', 'H20 ',       22.576D0,      548.900D0,00229200
     2       594.561D0,       .44505D0,      128.259D0,      423.970D0,00229300
     3    .18048935D+02,  .40960429D-01,  .42543344D-03, -.84894194D-06,00229400
     4    .74964999D-09, -.32490581D-12,  .55750386D-16/                00229500
C                                                                        00229600
      DATA    XPRP14          / 'N-DE', 'CANE', '    ', '    ', '    ',00229700
     1  '    ', '    ', 'N-C1', 'OH22',       20.693D0,      607.530D0,00229800
     2       617.550D0,       .48847D0,      142.287D0,      447.300D0,00229900
     3    .19702209D+02,  .46038560D-01,  .47657798D-03, -.96332701D-06,00230000
     4    .86409089D-09, -.38184805D-12,  .67103558D-16/                00230100
C                                                                        00230200
      DATA    XPRP15          / 'N-UN', 'DECA', 'NE  ', '    ', '    ',00230300
     1  '    ', '    ', 'N-C1', '1H24',       19.130D0,      665.000D0,00230400
     2       638.739D0,       .53014D0,      156.314D0,      469.080D0,00230500
```

```
     3     .21579239D+02,  .46998191D-01,  .55034064D-03, -.11325708D-05,00230600
     4     .10434666D-08, -.47541241D-12,  .86318577D-16/                 00230700
C                                                                           00230800
      DATA     XPRP16           / 'N-DO', 'DECA', 'NE  ', '    ', '    ',00230900
     1   '    ', '    ', 'N-C1', '2H26',       17.828D0,      719.700D0,00231000
     2        658.250D0,       .57083D0,       170.341D0,      489.470D0,00231100
     3     .17638199D+02,  .14817699D+00,  .12419743D-03, -.19432137D-06,00231200
     4    -.56242230D-12,  .90481892D-13, -.32622075D-16/                 00231300
C                                                                           00231400
      DATA     XPRP17           / 'N-TR', 'IDEC', 'ANE ', '    ', '    ',00231500
     1   '    ', '    ', 'N-C1', '3H28',       16.610D0,      775.200D0,00231600
     2        676.150D0,       .60960D0,       184.368D0,      508.620D0,00231700
     3     .28380463D+02,  .22925102D-08,  .92216274D-03, -.19254228D-05,00231800
     4     .18610851D-08, -.88915064D-12,  .16836658D-15/                 00231900
C                                                                           00232000
      DATA     XPRP18           / 'N-TE', 'TRAD', 'ECAN', 'E   ', '    ',00232100
     1   '    ', '    ', 'N-C1', '4H30',       15.525D0,      827.130D0,00232200
     2        692.950D0,       .64416D0,       198.395D0,      526.730D0,00232300
     3     .25255463D+02,  .85908607D-01,  .57805430D-03, -.11800856D-05,00232400
     4     .10398230D-08, -.44714033D-12,  .75924416D-16/                 00232500
C                                                                           00232600
      DATA     XPRP19           / 'N-PE', 'NTAD', 'ECAN', 'E   ', '    ',00232700
     1   '    ', '    ', 'N-C1', '5H32',       14.459D0,      880.280D0,00232800
     2        706.750D0,       .69180D0,       212.422D0,      543.870D0,00232900
     3     .22693863D+02,  .16252498D+00,  .27955425D-03, -.53815627D-06,00233000
     4     .33992350D-09, -.76462529D-13, -.42904145D-20/                 00233100
C                                                                           00233200
      DATA     XPRP20           / 'N-HE', 'XADE', 'CANE', '    ', '    ',00233300
     1   '    ', '    ', 'N-C1', '6H34',       13.576D0,      930.230D0,00233400
     2        720.550D0,       .73105D0,       226.449D0,      560.010D0,00233500
     3     .28160385D+02,  .10289364D+00,  .64807391D-03, -.13426743D-05,00233600
     4     .12019017D-08, -.52831352D-12,  .92412225D-16/                 00233700
C                                                                           00233800
      DATA     XPRP21           / 'N-HE', 'PTAD', 'ECAN', 'E   ', '    ',00233900
     1   '    ', '    ', 'N-C1', '7H36',       12.748D0,      977.000D0,00234000
     2        733.350D0,       .76233D0,       240.476D0,      575.200D0,00234100
     3     .29822159D+02,  .10787048D+00,  .69939681D-03, -.14565206D-05,00234200
     4     .13141384D-08, -.58277342D-12,  .10288245D-15/                 00234300
C                                                                           00234400
      DATA     XPRP22           / 'ETHY', 'LENE', '    ', '    ', '    ',00234500
     1   '    ', '    ', 'C2H4', '=   ',       49.700D0,      129.000D0,00234600
     2        282.400D0,       .08500D0,       28.054D0,       169.400D0,00234700
     3     .68128414D+01,  .17946490D-04,  .75724612D-04, -.11456202D-06,00234800
     4     .66020561D-10, -.87280118D-14, -.28854092D-17/                 00234900
C                                                                           00235000
      DATA     XPRP23           / 'PROP', 'YLEN', 'E   ', '    ', '    ',00235100
     1   '    ', '    ', 'C3H6', '=   ',       45.408D0,      181.000D0,00235200
     2        364.850D0,       .14400D0,       42.081D0,       225.460D0,00235300
     3     .79263134D+01,  .35798585D-05,  .14916311D-03, -.28064405D-06,00235400
     4     .24922234D-09, -.11089376D-12,  .19695749D-16/                 00235500
C                                                                           00235600
      DATA     XPRP24           / '1-BU', 'TENE', '    ', '    ', '    ',00235700
     1   '    ', '    ', '1-C4', 'H8  ',       39.700D0,      240.000D0,00235800
     2        419.600D0,       .18700D0,       56.108D0,       266.900D0,00235900
     3     .73262300D+01,  .35010204D-01,  .57559349D-04, -.74857452D-07,00236000
     4     .20407881D-11,  .30827962D-13, -.11133542D-16/                 00236100
C                                                                           00236200
      DATA     XPRP25           / 'CIS-', '2-BU', 'TENE', '    ', '    ',00236300
     1   '    ', '    ', 'C-2-', 'C4H8',       41.500D0,      234.000D0,00236400
     2        435.600D0,       .20200D0,       56.108D0,       276.900D0,00236500
     3     .96192379D+01,  .10515795D-07,  .17176652D-03, -.25410206D-06,00236600
     4     .15268957D-09, -.33898932D-13, -.19083062D-22/                 00236700
C                                                                           00236800
      DATA     XPRP26           / 'TRAN', 'S-2-', 'BUTE', 'NE  ', '    ',00236900
     1   '    ', '    ', 'T-2-', 'C4H8',       40.500D0,      238.000D0,00237000
     2        428.600D0,       .21400D0,       56.108D0,       274.000D0,00237100
     3     .88738861D+01,  .14174777D-01,  .16139694D-03, -.32203898D-06,00237200
     4     .29520414D-09, -.13642628D-12,  .25449097D-16/                 00237300
C                                                                           00237400
      DATA     XPRP27           / 'ISO-', 'BUTE', 'ENE ', '    ', '    ',00237500
     1   '    ', '    ', 'IC3-', '    ',       39.477D0,      239.000D0,00237600
     2        417.900D0,       .19400D0,       56.108D0,       266.200D0,00237700
```

```
     3    .58687611D+01,  .50494597D-01,  .18288603D-04, -.40314673D-07,00237800
     4   -.52767787D-12,  .20558363D-13, -.75344261D-17/                00237900
C                                                                       00238000
      DATA    XPRP28          / '1,3-', 'BUTA', 'DIEN', 'E   ', '    ',00238100
     1   '    ', '    ', '1,3-', 'C4==',       42.700DO,      221.000DO,00238200
     2        425.000DO,       .19500DO,       54.092DO,      268.700DO,00238300
     3    .46207895D+01,  .48677440D-01,  .19479788D-04, -.49883879D-07,00238400
     4    .11138439D-12,  .28137181D-13, -.10665350D-16/                00238500
C                                                                       00238600
      DATA    XPRP29          / '1-PE', 'NTEN', 'E   ', '    ', '    ',00238700
     1   '    ', '    ', '1-C5', 'H10 ',       40.000DO,      300.000DO,00238800
     2        464.700DO,       .24500DO,       70.135DO,      303.100DO,00238900
     3    .11449557D+02,  .87397409D-02,  .25250466D-03, -.49521402D-06,00239000
     4    .44436839D-09, -.19770821D-12,  .35071268D-16/                00239100
C                                                                       00239200
      DATA    XPRP30          / 'CIS-', '2-PE', 'NTEN', 'E   ', '    ',00239300
     1   '    ', '    ', 'C-2-', 'C5= ',       36.000DO,      300.000DO,00239400
     2        476.000DO,       .24000DO,       70.135DO,      310.100DO,00239500
     3    .10886964D+02,  .18074699D-03,  .26631192D-03, -.47132444D-06,00239600
     4    .37648896D-09, -.14593591D-12,  .22066633D-16/                00239700
C                                                                       00239800
      DATA    XPRP31          / 'TRAN', 'S-2-', 'PENT', 'ENE ', '    ',00239900
     1   '    ', '    ', 'T-2-', 'C5= ',       36.100DO,      300.000DO,00240000
     2        475.000DO,       .23700DO,       70.135DO,      309.500DO,00240100
     3    .10664039D+02,  .73196024D-02,  .27286010D-03, -.55874988D-06,00240200
     4    .53005516D-09, -.25088630D-12,  .47520815D-16/                00240300
C                                                                       00240400
      DATA    XPRP32          / '2-ME', 'THYL', '-1-B', 'UTEN', 'E   ',00240500
     1   '    ', '    ', '2MT-', '1C4=',       34.000DO,      318.000DO,00240600
     2        470.000DO,       .28500DO,       70.135DO,      311.700DO,00240700
     3    .71484098D+01,  .54705301D-01,  .81847484D-04, -.17371399D-06,00240800
     4    .11730191D-09, -.28000072D-13,  .43375310D-19/                00240900
C                                                                       00241000
      DATA    XPRP33          / '3-ME', 'THYL', '-1-B', 'UTEN', 'E   ',00241100
     1   '    ', '    ', '3MT-', '1C4=',       34.700DO,      300.000DO,00241200
     2        450.000DO,       .20900DO,       70.135DO,      293.300DO,00241300
     3    .45379162D+01,  .92181863D-01, -.31553563D-04, -.27930456D-07,00241400
     4    .31632638D-10, -.90871054D-14,  .10619723D-20/                00241500
C                                                                       00241600
      DATA    XPRP34          / '2-ME', 'THYL', '-2-B', 'UTEN', 'E   ',00241700
     1   '    ', '    ', '2MT-', '2C4=',       34.000DO,      318.000DO,00241800
     2        470.000DO,       .28500DO,       70.135DO,      311.700DO,00241900
     3    .96201916D+01,  .15760271D-01,  .21993817D-03, -.42323440D-06,00242000
     4    .36676477D-09, -.15714517D-12,  .26823430D-16/                00242100
C                                                                       00242200
      DATA    XPRP35          / '1-HE', 'XENE', '    ', '    ', '    ',00242300
     1   '    ', '    ', 'C6H1', '2=  ',       31.300DO,      350.000DO,00242400
     2        504.000DO,       .28500DO,       84.162DO,      336.600DO,00242500
     3    .12924405D+02,  .15954369D-01,  .29468316D-03, -.59197697D-06,00242600
     4    .54032622D-09, -.24432103D-12,  .43999125D-16/                00242700
C                                                                       00242800
      DATA    XPRP36          / 'CYCL', 'OPEN', 'TANE', '    ', '    ',00242900
     1   '    ', '    ', 'CYC-', 'C5  ',       44.491DO,      260.000DO,00243000
     2        511.700DO,       .19600DO,       70.140DO,      322.400DO,00243100
     3    .68700514D+01,  .42027257D-03,  .23024765D-03, -.28467266D-06,00243200
     4    .91958409D-10,  .34111618D-13, -.19240855D-16/                00243300
C                                                                       00243400
      DATA    XPRP37          / 'METH', 'YLCY', 'CLOP', 'ENTA', 'NE  ',00243500
     1   '    ', '    ', 'MTCY', 'C-C6',       37.345DO,      319.000DO,00243600
     2        532.730DO,       .23100DO,       84.160DO,      344.950DO,00243700
     3    .13644801D+02, -.74725459D-01,  .69186318D-03, -.13116369D-05,00243800
     4    .12042006D-08, -.55158466D-12,  .10065042D-15/                00243900
C                                                                       00244000
      DATA    XPRP38          / 'CYCL', 'OHEX', 'ANE ', '    ', '    ',00244100
     1   '    ', '    ', 'CYC-', 'C6  ',       40.168DO,      308.000DO,00244200
     2        553.500DO,       .21200DO,       84.160DO,      353.880DO,00244300
     3    .70494375D+01,  .80017628D-04,  .33824867D-03, -.50896443D-06,00244400
     4    .31352284D-09, -.77854579D-13,  .37046979D-17/                00244500
C                                                                       00244600
      DATA    XPRP39          / 'METH', 'YLCY', 'CLOH', 'EXAN', 'E   ',00244700
     1   '    ', '    ', 'MTCY', 'C-C6',       34.256DO,      368.000DO,00244800
     2        572.200DO,       .23600DO,       98.190DO,      374.080DO,00244900
```

```
     3    .83217106D+01, -.11802893D-02,  .47348840D-03, -.84093213D-06,00245000
     4    .65936050D-09, -.24415924D-12,  .33736263D-16/                 00245100
C                                                                       00245200
      DATA    XPRP40          / 'BENZ', 'ENE ', '    ', '    ', '    ',00245300
     1   '    ', '    ', 'BZ  ', '    ',       48.340D0,      259.000D0,00245400
     2        562.160D0,       .21200D0,       78.110D0,      353.240D0,00245500
     3    .36751518D+01,  .38030224D-01,  .95355250D-04, -.12393250D-06,00245600
     4    .77250878D-12,  .56966582D-13, -.20768997D-16/                 00245700
C                                                                       00245800
      DATA    XPRP41          / 'TOLU', 'ENE ', '    ', '    ', '    ',00245900
     1   '    ', '    ', 'TOL ', '    ',       40.503D0,      316.000D0,00246000
     2        591.790D0,       .26300D0,       92.140D0,      383.780D0,00246100
     3    .75503044D+01,  .12938867D-02,  .34966311D-03, -.68299765D-06,00246200
     4    .60172836D-09, -.25926373D-12,  .44180039D-16/                 00246300
C                                                                       00246400
      DATA    XPRP42          / 'ORTH', 'O-XY', 'LENE', '    ', '    ',00246500
     1   '    ', '    ', 'O-X ', '    ',       36.812D0,      369.000D0,00246600
     2        630.300D0,       .31000D0,      106.170D0,      417.580D0,00246700
     3    .81559858D+01,  .36442048D-01,  .27996444D-03, -.59909698D-06,00246800
     4    .55648350D-09, -.25352318D-12,  .45943105D-16/                 00246900
C                                                                       00247000
      DATA    XPRP43          / 'META', '-XYL', 'ENE ', '    ', '    ',00247100
     1   '    ', '    ', 'M-X ', '    ',       34.888D0,      376.000D0,00247200
     2        617.050D0,       .32500D0,      106.170D0,      412.270D0,00247300
     3    .90353098D+01,  .10930140D-01,  .38364207D-03, -.78521726D-06,00247400
     4    .73026082D-09, -.33607092D-12,  .61700011D-16/                 00247500
C                                                                       00247600
      DATA    XPRP44          / 'PARA', '-XYL', 'ENE ', '    ', '    ',00247700
     1   '    ', '    ', 'P-X ', '    ',       34.651D0,      379.000D0,00247800
     2        616.200D0,       .32100D0,      106.170D0,      411.520D0,00247900
     3    .95711050D+01,  .78086095D-02,  .38177642D-03, -.76441862D-06,00248000
     4    .69783379D-09, -.31544447D-12,  .56916441D-16/                 00248100
C                                                                       00248200
      DATA    XPRP45          / 'ETHY', 'LBEN', 'ZENE', '    ', '    ',00248300
     1   '    ', '    ', 'EB  ', '    ',       35.529D0,      374.000D0,00248400
     2        617.200D0,       .30200D0,      106.170D0,      409.340D0,00248500
     3    .94558764D+01,  .84744814D-04,  .44747164D-03, -.90994589D-06,00248600
     4    .83863563D-09, -.37872397D-12,  .67718502D-16/                 00248700
C                                                                       00248800
      DATA    XPRP46          / 'NITR', 'OGEN', '    ', '    ', '    ',00248900
     1   '    ', '    ', 'N2  ', '    ',       33.457D0,       89.000D0,00249000
     2        126.200D0,       .03900D0,       28.010D0,       77.350D0,00249100
     3    .69039288D+01,  .10369611D-11,  .58071192D-06,  .11181148D-08,00249200
     4   -.10532056D-11,  .22631708D-15, -.23281314D-26/                 00249300
C                                                                       00249400
      DATA    XPRP47          / 'OXYG', 'EN  ', '    ', '    ', '    ',00249500
     1   '    ', '    ', 'O2  ', '    ',       49.771D0,       73.000D0,00249600
     2        154.580D0,       .02400D0,       32.000D0,       90.180D0,00249700
     3    .67519922D+01,  .75807436D-03,  .24461570D-05, -.25044950D-08,00249800
     4    .10016526D-11, -.18067817D-15,  .12183202D-19/                 00249900
C                                                                       00250000
      DATA    XPRP48          / 'CARB', 'ON M', 'ONOX', 'IDE ', '    ',00250100
     1   '    ', '    ', 'CO  ', '    ',       34.532D0,       93.000D0,00250200
     2        132.910D0,       .05200D0,       28.010D0,       81.650D0,00250300
     3    .67485971D+01,  .74466500D-03,  .72321240D-06, -.37141838D-09,00250400
     4    .31753748D-14,  .20581126D-16, -.25343692D-20/                 00250500
C                                                                       00250600
      DATA    XPRP49          / 'CARB', 'ON D', 'IOXI', 'DE  ', '    ',00250700
     1   '    ', '    ', 'CO2 ', '    ',       72.860D0,       94.430D0,00250800
     2        304.210D0,       .22510D0,       44.010D0,      194.600D0,00250900
     3    .56366806D+01,  .14155747D-01, -.99279448D-05,  .38773364D-08,00251000
     4   -.84820125D-12,  .97181323D-16, -.45337256D-20/                 00251100
C                                                                       00251200
      DATA    XPRP50          / 'HYDR', 'OGEN', ' SUL', 'PHID', 'E   ',00251300
     1   '    ', '    ', 'H2S ', '    ',       88.231D0,       98.000D0,00251400
     2        373.200D0,       .10900D0,       34.080D0,      213.500D0,00251500
     3    .78297129D+01,  .64258499D-10,  .48952396D-05,  .37981427D-09,00251600
     4   -.38237442D-11,  .20021084D-14, -.31539707D-18/                 00251700
C                                                                       00251800
      DATA    XPRP51          / 'SULP', 'HUR ', 'DIOX', 'IDE ', '    ',00251900
     1   '    ', '    ', 'SO2 ', '    ',       77.809D0,      122.000D0,00252000
     2        430.800D0,       .25700D0,       64.060D0,      263.200D0,00252100
```

```
     3    .75942478D+01,  .33130160D-02,  .19711439D-04, -.35434399D-07,00252200
     4    .24974158D-10, -.81159935D-14,  .10072061D-17/                 00252300
C                                                                       00252400
      DATA    XPRP52           / '2-ME', 'THYL', '-PEN', 'TANE', '    ',00252500
     1   '    ', '    ', '2-MT', '-C5 ',      29.706D0,      367.000D0,00252600
     2        497.500D0,       .27800D0,       86.178D0,      333.410D0,00252700
     3    .71860952D+01,  .75717727D-01,  .10250425D-03, -.16853579D-06,00252800
     4   -.25632874D-12,  .11153552D-12, -.48404655D-16/                 00252900
C                                                                       00253000
      DATA    XPRP53           / '3-ME', 'THYL', 'PENT', 'ANE ', '    ',00253100
     1   '    ', '    ', '3-MT', '-C5 ',      30.831D0,      367.000D0,00253200
     2        504.500D0,       .27300D0,       86.178D0,      336.420D0,00253300
     3    .89448557D+01,  .91643076D-01, -.14514763D-04,  .15734975D-07,00253400
     4   -.29889279D-10, -.61427879D-18,  .64139897D-17/                 00253500
C                                                                       00253600
      DATA    XPRP54           / '2,2-', 'DIME', 'THYL', 'BUTA', 'NE  ',00253700
     1   '    ', '    ', '2,2D', 'MTC4',      30.397D0,      359.000D0,00253800
     2        488.780D0,       .23200D0,       86.178D0,      322.880D0,00253900
     3    .52148046D+01,  .96768749D-01,  .11150945D-04, -.18659698D-07,00254000
     4   -.34860062D-11, -.82176408D-13,  .72372053D-16/                 00254100
C                                                                       00254200
      DATA    XPRP55           / '2,3-', 'DIME', 'THYL', 'BUTA', 'NE  ',00254300
     1   '    ', '    ', '2,3D', 'MTC4',      30.861D0,      358.000D0,00254400
     2        499.980D0,       .24700D0,       86.178D0,      331.130D0,00254500
     3    .64690800D+01,  .82804874D-01,  .46561817D-04, -.45005110D-07,00254600
     4   -.38194312D-10, -.75539658D-14,  .34978877D-16/                 00254700
C                                                                       00254800
      DATA    XPRP56           / '1-HE', 'PTEN', 'E   ', '    ', '    ',00254900
     1   '    ', '    ', '1-C7', 'H14=',      28.000D0,      440.000D0,00255000
     2        537.200D0,       .35800D0,       98.189D0,      366.800D0,00255100
     3    .11443281D+02,  .75199352D-01,  .75832570D-04, -.11000747D-06,00255200
     4   -.18685235D-11,  .51822362D-13, -.18519920D-16/                 00255300
C                                                                       00255400
      DATA    XPRP57           / 'PROP', 'ADIE', 'NE  ', '    ', '    ',00255500
     1   '    ', '    ', 'C3H4', '==  ',      54.000D0,      162.000D0,00255600
     2        393.000D0,       .31300D0,       40.070D0,      238.700D0,00255700
     3    .73395252D+01,  .25859353D-02,  .12880163D-03, -.26785793D-06,00255800
     4    .25164929D-09, -.11538423D-12,  .20834873D-16/                 00255900
C                                                                       00256000
      DATA    XPRP58           / '1,2-', 'BUTA', 'DIEN', 'E   ', '    ',00256100
     1   '    ', '    ', '1,2-', 'C4==',      44.400D0,      219.000D0,00256200
     2        443.700D0,       .25500D0,       54.092D0,      284.000D0,00256300
     3    .61028252D+01,  .46325894D-01, -.51054392D-07, -.16761679D-07,00256400
     4   -.11230577D-11,  .79971874D-14, -.24261265D-17/                 00256500
C                                                                       00256600
      DATA    XPRP59           / 'ETHY', 'LCYC', 'LOPE', 'NTAN', 'E   ',00256700
     1   '    ', '    ', 'ETCY', 'C-C5',      33.526D0,      375.000D0,00256800
     2        569.500D0,       .27100D0,       98.190D0,      376.620D0,00256900
     3    .91773872D+01,  .39288596D-01,  .19385635D-03, -.21636194D-06,00257000
     4    .19215759D-11,  .88362908D-13, -.31171836D-16/                 00257100
C                                                                       00257200
      DATA    XPRP60           / 'ETHY', 'LCYC', 'LOHE', 'XANE', '    ',00257300
     1   '    ', '    ', 'ETCY', 'C-C6',      29.900D0,      450.000D0,00257400
     2        609.000D0,       .24300D0,      112.216D0,      404.900D0,00257500
     3    .91781473D+01,  .21337048D-02,  .57002378D-03, -.11023450D-05,00257600
     4    .97616516D-09, -.43011162D-12,  .76199984D-16/                 00257700
C                                                                       00257800
      DATA    XPRP61           / 'WATE', 'R   ', '    ', '    ', '    ',00257900
     1   '    ', '    ', 'H2O ', '    ',     217.617D0,       56.000D0,00258000
     2        647.140D0,       .32900D0,       18.020D0,      373.150D0,00258100
     3    .78004847D+01,  .36081475D-05,  .38989103D-05, -.24559562D-08,00258200
     4    .67111619D-12, -.87348723D-16,  .44003939D-20/                 00258300
C                                                                       00258400
      END                                                               00258500
C-----------------------------------------------------------------------C 00258600
      SUBROUTINE IOCHK                                                  00258700
C                                                                       00258800
C*********************************************************************** 00258900
C                                                                       00259000
C   THIS SUBROUTINE CHECKS THE PROCESS MATRIX TO SEE THAT CONSTRAINTS   00259100
C   ON THE NUMBER OF STREAMS ASSOCIATED WITH A UNIT ARE MET.            00259200
C                                                                       00259300
```

```
C*********************************************************************00259400
C                                                                     00259500
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)  00259600
     *           ,NMRCTR                                              00259700
      COMMON /CHARA/MODULE(5),CHRCHK(39),SEP(3),CHRUID(5)             00259800
      COMMON /INOUT/NW,NR                                             00259900
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)      00260000
     *             , ITRS,ITRT,MAXITR,CTOL                            00260100
      INTEGER FLGERR,UNTFLG                                           00260200
      CHARACTER MODULE*8,SEP,CHRCHK,CHRUID*2                          00260300
      DIMENSION NOUT(5),NIN(5)                                        00260400
      DO 12 I=1,MAXUID                                                00260500
      IF(NPM(I,2) .NE. 1 .AND. NPM(I,2) .NE. 2) GOTO 12               00260700
         NUMIN=0                                                      00260800
         NUMOUT=0                                                     00260900
         CALL NOUTO(NOUT)                                             00261000
         CALL NOUTO(NIN)                                              00261100
         CALL FNDOUT(I,NOUT,NUMOUT)                                   00261200
         CALL FNDIN(I,NIN,NUMIN)                                      00261300
         NERF=NPM(I,2)                                                00261400
         GOTO (13,26),NERF                                            00261500
C                                                                     00261600
C   CHECK MIXER                                                       00261700
C                                                                     00261800
13       MAXIN=4                                                      00261900
         MAXOUT=1                                                     00262000
         MINOUT=1                                                     00262100
         MININ=2                                                      00262200
         IF(NUMIN .LT. MININ) THEN                                    00262400
            WRITE(NW,116)                                             00262500
116         FORMAT(' ****  ERROR IN PROCESS TOPOLOGY  ****'/)         00262600
            WRITE(NW,112) I,MININ,MODULE(NPM(I,2))                    00262700
112         FORMAT(' THE MINIMUM NUMBER OF INLET STREAMS TO UNIT',    00262800
     *           ' #',I2,' HAS NOT BEEN MET.'/' THERE MUST BE AT',    00262900
     *           ' LEAST ',I1,' STREAMS TO A ',A8/)                   00263000
            WRITE(NW,117)                                             00263100
117         FORMAT(' !!!!  PROCESS TOPOLOGY MUST NOW BE RESET  !!!!') 00263200
            GOTO 14                                                   00263300
         END IF                                                       00263400
         IF(NUMIN .GT. MAXIN) THEN                                    00263500
            WRITE(NW,116)                                             00263600
            WRITE(NW,113)  I,MAXIN,MODULE(NPM(I,2))                   00263700
113         FORMAT(' THE MAXIMUM NUMBER OF INLET STREAMS TO UNIT #',  00263800
     *           I1,' HAS BEEN EXCEEDED.'/' THERE CAN BE ONLY ',I1,   00263900
     *           ' STREAMS TO A ',A8/)                                00264000
            WRITE(NW,117)                                             00264100
            GOTO 14                                                   00264200
         END IF                                                       00264300
         IF(NUMOUT .LT. MINOUT) THEN                                  00264400
            WRITE(NW,116)                                             00264500
            WRITE(NW,118) I,MINOUT,MODULE(NPM(I,2))                   00264600
118         FORMAT(' THE MINIMUM NUMBER OF OUTLET STREAMS FROM UNIT #'00264700
     *           ,I1,' HAS NOT BEEN MET.'/' THERE MUST BE AT LEAST',I1,00264800
     *           ' STREAMS FROM A ',A8/)                              00264900
            WRITE(NW,117)                                             00265000
            GOTO 14                                                   00265100
         END IF                                                       00265200
         IF(NUMOUT .GT. MAXOUT) THEN                                  00265300
            WRITE(NW,116)                                             00265400
            WRITE(NW,119) I,MAXOUT,MODULE(NPM(I,2))                   00265500
119         FORMAT(' THE MAXIMUM NUMBER OF OUTLET STREAMS FROM UNIT #'00265600
     *           ,I1,' HAS BEEN EXCEEDED.'/' THERE CAN BE ONLY ',I1,  00265700
     *           ' STREAMS FROM A ',A8/)                              00265800
            WRITE(NW,117)                                             00265900
            GOTO 14                                                   00266000
         END IF                                                       00266100
      GOTO 12                                                         00266200
C                                                                     00266300
C   CHECK SPLITTER                                                    00266400
C                                                                     00266500
26    MAXIN=1                                                         00266600
      MAXOUT=4                                                        00266700
```

```
      MININ=1                                                          00266800
      MINOUT=1                                                         00266900
      IF(NUMIN .LT. MININ) THEN                                        00267000
         WRITE(NW,116)                                                 00267100
         WRITE(NW,112) I,MININ,MODULE                                  00267200
         WRITE(NW,117)                                                 00267300
         GOTO 14                                                       00267400
      END IF                                                           00267500
      IF(NUMIN .GT. MAXIN) THEN                                        00267600
         WRITE(NW,116)                                                 00267700
         WRITE(NW,113) I,MAXIN,MODULE(NPM(I,2))                        00267800
         WRITE(NW,117)                                                 00267900
         GOTO 14                                                       00268000
      END IF                                                           00268100
      IF(NUMOUT .LT. 1) THEN                                           00268200
         WRITE(NW,116)                                                 00268300
         WRITE(NW,118) I,MINOUT,MODULE(NPM(I,2))                       00268400
         WRITE(NW,117)                                                 00268500
         GOTO 14                                                       00268600
      END IF                                                           00268700
      IF(NUMOUT .GT. 4) THEN                                           00268800
         WRITE(NW,116)                                                 00268900
         WRITE(NW,119) I,MAXOUT,MODULE(NPM(I,2))                       00269000
         WRITE(NW,117)                                                 00269100
         GOTO 14                                                       00269200
      END IF                                                           00269300
      FLGERR=0                                                         00269400
12    CONTINUE                                                         00269500
C                                                                      00269600
C   NOW CHECK FOR ONE INPUT AND ONE OUTPUT FOR CSTR AND PFR.           00269700
C                                                                      00269800
      MINOUT=1                                                         00269900
      MAXOUT=1                                                         00270000
      MAXIN=1                                                          00270100
      MAXOUT=1                                                         00270200
      DO 15 I=1,5                                                      00270300
         NUMIN=0                                                       00270400
         NUMOUT=0                                                      00270500
         IF(NPM(I,2)-3) 15,16,16                                       00270600
16       CALL NOUTO(NIN)                                               00270700
         CALL FNDIN(I,NIN,NUMIN)                                       00270800
         CALL NOUTO(NOUT)                                              00270900
         CALL FNDOUT(I,NOUT,NUMOUT)                                    00271000
         IF(NUMIN .GT. MAXIN) THEN                                     00271100
            WRITE(NW,116)                                              00271200
            WRITE(NW,113)  I,MAXIN,MODULE(NPM(I,2))                    00271300
            WRITE(NW,117)                                              00271400
            GOTO 14                                                    00271500
         END IF                                                        00271600
         IF(NUMIN .LT. 0) THEN                                         00271700
            WRITE(NW,116)                                              00271800
            WRITE(NW,112) I,MININ,MODULE(NPM(I,2))                     00271900
            WRITE(NW,117)                                              00272000
            GOTO 14                                                    00272100
         END IF                                                        00272200
         IF(NUMOUT .GT. MAXOUT) THEN                                   00272300
            WRITE(NW,116)                                              00272400
            WRITE(NW,119) I,MAXOUT,MODULE(NPM(I,2))                    00272500
            WRITE(NW,117)                                              00272600
            GOTO 14                                                    00272700
         END IF                                                        00272800
         IF(NUMOUT .LT. MINOUT) THEN                                   00272900
            WRITE(NW,116)                                              00273000
            WRITE(NW,118) I,MINOUT,MODULE(NPM(I,2))                    00273100
            WRITE(NW,117)                                              00273200
            GOTO 14                                                    00273300
         END IF                                                        00273400
15    CONTINUE                                                         00273500
      RETURN                                                           00273600
14    CALL INISHL                                                      00273700
      FLGERR=1                                                         00273800
C                                                                      00273900
```

```
C    FLGERR= 0    NO ERROR IN PROCESS (OR SUBROUTINE HAS BEEN CHECKED      00274000
C                 AND IS FOUND TO BE ERRORLESS).                           00274100
C    FLGERR= 1    AN ERROR HAS BEEN FOUND IN THE SUBROUTINE OR PROCESS.    00274200
C                                                                          00274300
      RETURN                                                               00274400
      END                                                                  00274500
      SUBROUTINE FNDIN(I,NIN,NUMIN)                                        00274600
C                                                                          00274700
C*************************************************************************00274800
C                                                                          00274900
C    THE PURPOSE OF THIS ROUTINE IS TO FIND THE NUMBER OF INLET STREAMS    00275000
C    TO A UNIT OPERATION AND PUT THE STREAM NUMBERS INTO AN ARRAY.         00275100
C                                                                          00275200
C*************************************************************************00275300
C                                                                          00275400
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)       00275500
     *           ,NMRCTR                                                   00275600
      DIMENSION NIN(5)                                                     00275700
      NUMIN=0                                                              00275800
      K=1                                                                  00275900
      DO 10 J=3,7                                                          00276000
         IF(NPM(I,J) .LE. 0) GOTO 10                                       00276100
         NUMIN=NUMIN+1                                                     00276200
         NIN(K)=NPM(I,J)                                                   00276300
         K=K+1                                                             00276400
10    CONTINUE                                                             00276500
      NUMIN=K-1                                                            00276600
      RETURN                                                               00276700
      END                                                                  00276800
      SUBROUTINE FNDNRT(NIN)                                               00276900
C                                                                          00277000
C*************************************************************************00277100
C                                                                          00277200
C    THE PUROPOSE OF THIS ROUTINE IS TO SEARCH THE STREAM(S) AS            00277300
C    SPECIFIED IN NIN FOR COMPONENTS THAT ARE NOT PRESENT IN               00277400
C    THE NCOEF MATRIX. THESE CMP ID #'S ARE THEN PLACED IN THE             00277500
C    INERT ARRAY. THEY ARE THUS TAGGED AS INERT CMPS.                      00277600
C                                                                          00277700
C    THIS ROUTINE ALSO CALCULATES THE TOTAL FLOW OF INERTS (FLONRT(6)) IN  00277800
C    THE NIN(1) (FEED TO A REACTOR) STREAM.                                00277900
C                                                                          00278000
C    SINCE THERE MAY BE MORE THAN ONE REACTOR IN THE PROCESS,              00278100
C    THERE MAY BE MORE THAN ONE SET OF INERTS IN A PROCESS FEED STREAM     00278200
C    TO A REACTOR. SO THIS ROUTINE SHOULD BE CALLED FOR EACH REACTOR       00278300
C    WHERE UNTFLG(INDX,1)=0.                                               00278400
C                                                                          00278500
C    INERT INITIALIZED TO 0.                                               00278600
C*************************************************************************00278700
C                                                                          00278800
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                   00278900
     *          ,IPHASE,MAXCMP,INERT(5)                                    00279000
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                      00279100
     *          ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS    00279200
     *          ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)        00279300
     *          ,RXN(5,14)                                                 00279400
      DOUBLE PRECISION FLONRT                                              00279500
      CHARACTER RXN                                                        00279600
      DIMENSION NIN(5)                                                     00279700
      K=0                                                                  00279800
      DO 12 I=1,6                                                          00279900
12    FLONRT(I)=0.                                                         00280000
CU    FLONRT{MOL/MIN}                                                      00280100
      DO 10 I=1,9                                                          00280200
         DO 11 J=1,8                                                       00280300
            IF(INT(STRMID(NIN(1),I,1)) .EQ. NCOEF(J,1) .OR.                00280400
     *      STRMID(NIN(1),I,1) .LT. 0) GOTO 10                             00280500
            IF(J .EQ. 8  ) THEN                                            00280600
               K=K+1                                                       00280700
               INERT(K)=INT(STRMID(NIN(1),I,1))                            00280800
               FLONRT(K)=STRMID(NIN(1),I,2)*1000.                          00280900
               FLONRT(6)=FLONRT(6) + STRMID(NIN(1),I,2)*1000.              00281000
            END IF                                                         00281200
```

```
11        CONTINUE                                                      00281300
10     CONTINUE                                                         00281400
       RETURN                                                           00281800
       END                                                              00281900
       SUBROUTINE STPOS                                                 00282000
C                                                                       00282100
C***********************************************************************00282200
C                                                                       00282300
C    THE PURPOSE OF THIS ROUTINE IS TO SET UP AN ARRAY (NPOSPC(7,5)  THAT00282400
C    HOLDS THE POSITIONS (ROW NUMBERS) OF THE REACTANTS AND PRODUCTS     00282500
C    OF ALL THE REACTIONS. NPOSPC IS FILLED SEQUENTIALLY.               00282600
C                                                                       00282700
C    EG.    IF NCOEF =     12  -1  0  0  0  0                          00282800
C                          10  -1  0  0  0  0                          00282900
C                           3   2 -1  0  0  0                          00283000
C                           6   0 -1  0  0  0                          00283100
C                           4   0  2  0  0  0                          00283200
C                                                                       00283300
C                                                                       00283400
C    THEN NPOSPC    =        1  3                                       00283500
C                            2  4                                       00283600
C                            3  5                                       00283700
C                            0  0                                       00283800
C                            0  0                                       00283900
C    # OF REACTANTS -->      2  2                                       00284000
C    # OF PRODUCTS  -->      1  1                                       00284100
C                                                                       00284200
C***********************************************************************00284300
C                                                                       00284400
C    VARIABLES:                                                         00284500
C                                                                       00284600
C           N = # OF REACTANTS IN ANY GIVEN REACTION                    00284700
C           L = # OF PRODUCTS IN ANY GIVEN REACTION                     00284800
C           K = A COUNTER THAT TRACKS THE ROW POSITION THAT THE         00284900
C               CURRENT VALUE OF NPOSPC IS TO OCCUPY                    00285000
C                                                                       00285100
C***********************************************************************00285200
C                                                                       00285300
       COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                  00285400
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS  00285500
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)      00285600
     *         ,RXN(5,14)                                               00285700
       COMMON /INOUT/NW,NR                                              00285800
       CHARACTER RXN                                                    00285900
C                                                                       00286000
C                                                                       00286100
C    INSERT THE POSITION OF THE REACTANTS                               00286200
C                                                                       00286300
       DO 10 J=2,6                                                      00286400
          K=0                                                           00286500
          N=0                                                           00286600
          L=0                                                           00286700
          DO 11 I=1,9                                                   00286800
             IF(NCOEF(I,J) .LT. 0) THEN                                 00286900
                N=N+1                                                   00287000
                K=K+1                                                   00287100
                NPOSPC(K,J-1)=I                                         00287200
             END IF                                                     00287300
             IF(NCOEF(I,J) .GT. 0) THEN                                 00287400
                L=L+1                                                   00287500
                K=K+1                                                   00287600
                NPOSPC(K,J-1)=I                                         00287700
             END IF                                                     00287800
11        CONTINUE                                                      00287900
          NPOSPC(6,J-1)=N                                               00288000
          NPOSPC(7,J-1)=L                                               00288100
10     CONTINUE                                                         00288200
       RETURN                                                           00288700
       END                                                              00288800
       SUBROUTINE STRK                                                  00288900
C                                                                       00289000
C***********************************************************************00289100
```

```
C                                                                       00289200
C   THE PURPOSE OF THIS ROUTINE IS TO REQUEST RATE CONSTANT DATA        00289210
C   FOR ISOTHERMAL REACTIONS.                                           00289220
C                                                                       00289230
C*********************************************************************  00289240
C                                                                       00289250
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                   00289300
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS  00289400
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00289500
     *         ,RXN(5,14)                                                00289600
      COMMON /INOUT/NW,NR                                               00289700
      CHARACTER RXN                                                     00289800
      DO 10 I=1,NRXNS                                                   00290000
         WRITE(NW,100) I                                                00290100
100      FORMAT(' ENTER RATE CONSTANT (L-MOLE-MIN) FOR FORWARD REACTION'00290200
     *          ,I2)                                                    00290300
         READ(NR,*) RK(I,1)                                             00290400
         IF(IDIR(I) .EQ. 1) THEN                                        00290500
            WRITE(NW,101) I                                             00290600
101         FORMAT(' ENTER EQUILIBRIUM CONSTANT (DIMENSIONLESS) FOR '   00290700
     *            ,'REACTION',I2)                                       00290800
            READ(NR,*) RK(I,2)                                          00290900
         END IF                                                         00291000
10    CONTINUE                                                          00291100
      RETURN                                                            00291700
      END                                                               00291800
C                                                                       00291900
C*********************************************************************  00292000
C                                                                       00292100
C   THE PURPOSE OF THIS ROUTINE IS TO SET THE EXPONENTS ON THE CON-     00292110
C   CENTRATION TERMS FROM THE COEFFICIENTS IN THE STOICHIOMETRIC.       00292120
C   EQUATIONS.                                                          00292130
C                                                                       00292140
C*********************************************************************  00292150
C                                                                       00292160
      SUBROUTINE STEXP                                                  00292200
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                   00292300
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS  00292400
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00292500
     *         ,RXN(5,14)                                                00292600
      COMMON /INOUT/NW,NR                                               00292700
      CHARACTER RXN                                                     00292800
      DO 12 J=2,6                                                       00293000
         DO 10 I=1,9                                                    00293100
            IF(EXP(I,J-1) .NE. 0) GOTO 10                               00293200
            EXP(I,J-1)=FLOAT(IABS(NCOEF(I,J)))                          00293300
10       CONTINUE                                                       00293400
12    CONTINUE                                                          00293500
      RETURN                                                            00294100
      END                                                               00294200
      SUBROUTINE STE                                                    00294300
C                                                                       00294400
C*********************************************************************  00294500
C                                                                       00294600
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                   00294700
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS  00294800
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00294900
     *         ,RXN(5,14)                                                00295000
      CHARACTER RXN                                                     00295100
      COMMON /INOUT/NW,NR                                               00295200
      DO 10 I=1,NRXNS                                                   00295400
         WRITE(NW,100) I                                                00295500
100      FORMAT(' ENTER ACTIVATION ENERGY (CAL/MOL) FOR '               00295600
     *          ,'REACTION ',I2)                                        00295700
         READ(NR,*) E(I)                                                00295800
10    CONTINUE                                                          00295900
      RETURN                                                            00296400
      END                                                               00296500
      SUBROUTINE STA                                                    00296600
C                                                                       00296700
C*********************************************************************  00296800
C                                                                       00296900
```

```
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                    00297000
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS   00297100
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00297200
     *         ,RXN(5,14)                                                00297300
      CHARACTER RXN                                                      00297400
      COMMON /INOUT/NW,NR                                                00297500
      DO 10 I=1,NRXNS                                                    00297600
         WRITE(NW,100) I                                                 00297700
100      FORMAT(' ENTER FREQUENCY FACTOR (L-MOL-MIN) FOR '               00297800
     *          ,'REACTION',I2)                                          00297900
         READ(NR,*) A(I)                                                 00298000
10    CONTINUE                                                           00298100
      RETURN                                                             00298600
      END                                                                00298700
      SUBROUTINE STDELH                                                  00298800
C                                                                        00298900
C***********************************************************************  00299000
C                                                                        00299100
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                    00299200
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS   00299300
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00299400
     *         ,RXN(5,14)                                                00299500
      CHARACTER RXN                                                      00299600
      COMMON /INOUT/NW,NR                                                00299700
      DO 10 I=1,NRXNS                                                    00299800
         WRITE(NW,100) I                                                 00299900
100      FORMAT(' ENTER HEAT OF REACTION (CAL/MOL) FOR REACTION ',I2)    00300000
10       READ(NR,*) DELH(I)                                              00300100
      RETURN                                                             00300200
      END                                                                00300300
      SUBROUTINE STX(INDX)                                               00300400
C                                                                        00300500
C***********************************************************************  00300600
C                                                                        00300700
C   THE PURPOSE OF THIS ROUTINE IS TO REQUEST CONVERSION FOR A           00300800
C   SPECIFIED REACTOR = INDX. THIS ROUTINE IS CALLED THE FIRST           00300900
C   TIME A CALL TO A REACTOR IS MADE.                                    00301000
C                                                                        00301100
C***********************************************************************00301200
C                                                                        00301300
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)     00301400
     *          ,NMRCTR                                                  00301500
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                    00301600
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS   00301700
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00301800
     *         ,RXN(5,14)                                                00301900
      COMMON /INOUT/NW,NR                                                00302000
      COMMON /CHARA/MODULE(5),CHRCHK(39),SEP(3),CHRUID(5)                00302100
      CHARACTER RXN,CHRCHK,SEP,MODULE*8,CHRUID*2                         00302200
      WRITE(NW,100) NPM(INDX,1),MODULE(NPM(INDX,2))                      00302300
      READ(NR,*) X(INDX)                                                 00302400
      CALL RERRLM(0.01,0.99,X(INDX))                                     00302410
100   FORMAT(' ENTER CONVERSION FOR UNIT # ',I2,'-',A10)                 00302500
      RETURN                                                             00302700
      END                                                                00302800
      SUBROUTINE FNDKEY                                                  00302900
C                                                                        00303000
C***********************************************************************  00303100
C                                                                        00303200
C   THIS ROUTINE FINDS THE KEY COMPONENT FOR EACH REACTION. THE KEY      00303300
C   COMPONENT IS DEFINED AS THE FIRST COMPNENT IN EACH REACTION.         00303400
C   THE RATE EXPRESSION IS THEN WRITTEN FOR THAT COMPONENT.              00303500
C                                                                        00303600
C***********************************************************************  00303700
C                                                                        00303710
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                    00303800
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS   00303900
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00304000
     *         ,RXN(5,14)                                                00304100
      COMMON /INOUT/NW,NR                                                00304200
      CHARACTER RXN                                                      00304300
      JJ=NRXNS+1                                                         00304400
```

```
      DO 10 J=2, JJ                                                    00304500
         DO 11 I=1,8                                                   00304600
            IF(NCOEF(I,J) .NE. 0) THEN                                 00304700
               KEYPOS(J-1)=I                                           00304800
               LL=J-1                                                  00304900
               GOTO 10                                                 00305100
            END IF                                                     00305200
11       CONTINUE                                                      00305300
10    CONTINUE                                                         00305400
      RETURN                                                           00305800
      END                                                              00305900
      SUBROUTINE STRMCV(INDX,STNUM,N)                                  00306000
C                                                                      00306100
C*********************************************************************00306200
C                                                                      00306300
C   THIS ROUTINE COMPARES NEWLY CALCULATED STREAMS TO PREVIOUSLY       00306400
C   CALCULATED STREAMS AND DETERMINES IF THE STREAMS HAVE CON-         00306500
C   VERGED. IF ALL THE INLET AND OUTLET STREAMS HAVE CONVERGED,        00306600
C   STRMCV SETS THE UNIT'S UNTFLG(INDX,2) = 1 (DON'T WANT THE          00306700
C   UNIT TO CALCULATED ANY MORE). THIS ROUTINE IS CALLED FROM          00306800
C   MIXER, SPLTTR, AND PFR ONLY IF RECYCLE STREAM(S) IS/ARE PRESENT.   00306900
C                                                                      00307000
C                                                                      00307100
C   VARIABLES:                                                         00307200
C                                                                      00307300
C         INDX = UNIT NUMBER THAT CALLED STRMCV. THIS IS FROM THE      00307400
C                FIRST COLUMN OF THE NPM MATRIX.                       00307500
C                                                                      00307600
C        STNUM = ARRAY THAT HOLDS THE OUTLET STREAM NUMBERS.           00307700
C                                                                      00307800
C            N = NUMBER OF OUTLET STREAM NUMBERS THAT STRMCV           00307900
C                IS CURRENTLY WORKING ON.                              00308000
C                                                                      00308100
C        STFLO = STREAM FLOW. HOLDS PREVIOUSLY CALCULATED STREAM       00308200
C                FLOWS (KG-MOLES/MIN).                                 00308300
C                                                                      00308400
C         IFLG = 0 = ALL TOLERANCES MET. STREAM(S) HAVE CONVERGED.     00308500
C              = 1 = AT LEAST ONE TOLERANCE HAS NOT BEEN MET.          00308600
C                                                                      00308700
C*********************************************************************00308800
C                                                                      00308900
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO               00309000
     *         ,IPHASE,MAXCMP,INERT(5)                                 00309100
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                     00309200
     *             CMP(100)                                            00309300
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)       00309400
     *             , ITRS,ITRT,MAXITR,CTOL                             00309500
      COMMON /INOUT/NW,NR                                              00309600
      DOUBLE PRECISION FLONRT                                          00309700
      INTEGER STNUM(5),UNTFLG,FLGERR                                   00309800
      DATA ITER/0/                                                     00309900
      IF(INDX .EQ. 1) ITER=ITER+1                                      00310000
      IFLG=0                                                           00310400
      I=1                                                              00310500
11    IF(STNUM(I) .NE. 0) THEN                                         00310600
         NS=STNUM(I)                                                   00310700
         J=1                                                           00310800
12       IF(STRMID(NS,J,2) .GT. 0.) THEN                               00310900
            TOL=ABS(STRMID(NS,J,2) - STFLO(NS,J))/STRMID(NS,J,2)       00311000
            STFLO(NS,J)=STRMID(NS,J,2)                                 00311100
            IF(TOL .GT. CTOL) IFLG=1                                   00311200
            J=J + 1                                                    00311300
            GOTO 12                                                    00311400
         END IF                                                        00311500
         I=I + 1                                                       00311600
         GOTO 11                                                       00311700
      END IF                                                           00311800
      IF(IFLG .EQ. 1) GOTO 10                                          00311900
      WRITE(NW,100) INDX                                               00312000
100   FORMAT(' >>>>>>>>>>>>>    UNIT # ',I1,' HAS CONVERGED'/)          00312100
      UNTFLG(INDX,2)=1                                                 00312200
      NFLGCV(INDX)=0                                                   00312300
```

```
10    RETURN                                                            00312400
      END                                                               00312500
      SUBROUTINE STRCT(INDX,NIN)                                        00312600
C                                                                       00312700
C***********************************************************************00312800
C                                                                       00312900
C   THE PURPOSE OF THIS ROUTINE IS TO TRANSFER THE DATA FROM            00313000
C   THE STRMID ARRAY FOR THE STREAM THAT IS THE FEED TO A REACTOR       00313100
C   TO THE ARRAY RCT. THE RCT ARRAY WILL HOLD THE FOLLOWING             00313200
C   INFORMATION:                                                        00313300
C                                                                       00313400
C   COLUMNS = ITERATION # ;  F_ ARE THE FLOWS OF THE CMPS.              00313500
C   ROWS ARE AS FOLLOWS,                                                00313600
C                                                                       00313700
C    F1  F2  F3  F4  F5  F6  F7  F8  F9  TOTF  XA   V   T(K) 1/(-RA)    00313800
C                                                                       00313900
C   THE FIRST ROW WILL HOLD THE CMP ID #'S. THESE NUMBERS ARE           00314000
C   INSERTED SEQUENTIALLY FROM THE NCOEF ARRAY. ONLY REACTIVE CMPS      00314100
C   ARE CARRIED IN RCT. IF THERE ARE INERTS PRESENT, THESE FLOWS        00314200
C   ARE CARRIED IN THE INERT ARRAY, AS THESE WILL NOT CHANGE.           00314300
C   THE ARRAY RCT IS INITIALIZED TO 0.                                  00314400
C                                                                       00314500
C   THE FIRST SUBSCRIPT REFERS TO THE INDX NUMBER FOR THE REACTOR       00314600
C   BEING CALCULATED.                                                   00314700
C                                                                       00314800
C***********************************************************************00314900
C                                                                       00315000
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7) ,FAO,P,VFLO               00315100
     *         ,IPHASE,MAXCMP,INERT(5)                                  00315200
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00315300
     *           ,NMRCTR                                                00315400
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                   00315500
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS  00315600
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)      00315700
     *         ,RXN(5,14)                                               00315800
      COMMON /INOUT/NW,NR                                               00315900
      DOUBLE PRECISION FLONRT                                           00316000
      CHARACTER RXN                                                     00316100
      DIMENSION NIN(5)                                                  00316200
      IF(STRMID(NIN(1),10,1) .LE. 0. .AND. (IPHASE .EQ. 1 .OR.          00316700
     *  NOPCND(INDX) .EQ. 2)) THEN                                      00316800
          CALL STTEMP(INDX)                                             00316900
C                                                                       00317000
C  INSERT TOTAL MOLE (KG-MOL/MIN) INTO RCT(2,10)                        00317100
C                                                                       00317200
      END IF                                                            00317400
C                                                                       00317500
C   INSERT CMP ID #'S                                                   00317600
C                                                                       00317700
      DO 11 I=1,9                                                       00317800
         IF(NCOEF(I,1) .LE. 0) GOTO 12                                  00317900
11       RCT(INDX,1,I)=FLOAT(NCOEF(I,1))                                00318000
C                                                                       00318100
C   FIND ALL THE COMPONENTS THAT ARE PRESENT IN STRMID AND RCT.         00318200
C   INSERT THEIR FLOWS.                                                 00318300
C   CALCULATE TOTAL MOLES THEN INSERT INTO RCT.                         00318400
C                                                                       00318500
12    SUM=0.                                                            00318600
      DO 13 I=1,9                                                       00318700
         IF(RCT(INDX,1,I) .EQ. 0.) GOTO 15                              00318800
         DO 14 J=1,9                                                    00318900
            IF(RCT(INDX,1,I) .EQ. STRMID(NIN(1),J,1)) THEN              00319000
               RCT(INDX,2,I)=STRMID(NIN(1),J,2)                         00319100
               SUM=SUM+RCT(INDX,2,I)                                    00319200
               GOTO 13                                                  00319300
            END IF                                                      00319400
14       CONTINUE                                                       00319500
13    CONTINUE                                                          00319600
15    RCT(INDX,2,10)=SUM                                                00319700
      RCT(INDX,2,13)=STRMID(NIN(1),10,1)                                00320000
      RCT(INDX,2,10)=RCT(INDX,2,10) + FLONRT(6)/1000.                   00320100
      RETURN                                                            00320700
```

```
      END                                                                00320800
      SUBROUTINE PFR(INDX)                                               00320900
C                                                                        00321000
C***********************************************************************00321100
C                                                                        00321200
C   ONLY ONE INLET AND ONE OUTLET STREAM FOR ANY REACTOR MODULE.         00321300
C                                                                        00321400
C   THE ROUTINE STRCT WILL CALL STEMP                                    00321500
C                                                                        00321510
C***********************************************************************00321520
C                                                                        00321530
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7) ,FAO,P,VFLO                00321600
     *         ,IPHASE,MAXCMP,INERT(5)                                   00321700
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                       00321800
     *            CMP(100)                                               00321900
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)     00322000
     *           ,NMRCTR                                                 00322100
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                    00322200
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS   00322300
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00322400
     *         ,RXN(5,14)                                                00322500
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)         00322600
     *              , ITRS,ITRT,MAXITR,CTOL                              00322700
      COMMON /INOUT/NW,NR                                                00322800
      DOUBLE PRECISION FLONRT                                            00322900
      CHARACTER RXN                                                      00323000
      DIMENSION NIN(5),NOUT(5)                                           00323100
      DO 10 I=1,100                                                      00323400
10       CMP(I)=0.                                                       00323500
      CALL NOUTO(NOUT)                                                   00323600
      CALL NOUTO(NIN)                                                    00323700
      CALL FNDOUT(INDX,NOUT,NUMOUT)                                      00323800
      CALL FNDIN(INDX,NIN,NUMIN)                                         00323900
      IF(IPHASE .EQ. 1) CALL FNDNRT(NIN)                                 00324000
CU    FAO{MOL/MIN}                                                       00324300
      CALL STRCT(INDX,NIN)                                               00325700
      FAO=RCT(INDX,2,1)*1.D3                                             00325800
CU    RCT FLOWS{KG-MOL/MIN}                                              00325900
CU    FAO{MOL/MIN}                                                       00326000
      CALL STPOS                                                         00326100
      CALL TFACE(INDX,NIN,NUMIN)                                         00326200
      CALL ODE(INDX,NIN,NUMIN,NOUT,NUMOUT)                               00326300
      IF(NFLGCV(INDX) .EQ. 1) CALL STRMCV(INDX,NOUT,NUMOUT)              00326400
      CALL TFACE(INDX,NOUT,NUMOUT)                                       00326500
      UNTFLG(INDX,1)=1                                                   00328500
      RETURN                                                             00328600
      END                                                                00328700
      SUBROUTINE CSTR(INDX)                                              00328800
C                                                                        00328900
C***********************************************************************00329000
C                                                                        00329100
      COMMON /INOUT/NW,NR                                                00329200
      WRITE(NW,100)                                                      00329300
100   FORMAT(' CSTR UNIT OPERATION IS NOT FUNCTIONAL.')                  00329400
      RETURN                                                             00329500
      END                                                                00329600
      SUBROUTINE BATCH(INDX)                                             00329700
C                                                                        00329800
C***********************************************************************00329900
C                                                                        00330000
      COMMON /INOUT/NW,NR                                                00330100
      WRITE(NW,100)                                                      00330200
100   FORMAT(' BATCH UNIT OPERATION IS NOT FUNCTIONAL.')                 00330300
      RETURN                                                             00330400
      END                                                                00330500
      SUBROUTINE ODE(INDX,NIN,NUMIN,NOUT,NUMOUT)                         00330600
C                                                                        00330700
C***********************************************************************00330800
C                                                                        00330900
C   ODE USES THE LSODAR ROUTINE FROM THE DIFFERENTIAL EQUATION           00331000
C   SOLVING PACKAGE ODEPACK, WRITTEN BY HINDMARSH AND PECTOLD            00331100
C   AT THE LAWRENCE LIVERMORE LABORATORY.                                00331200
```

```
C                                                                       00331300
C*********************************************************************  00331400
C                                                                       00331500
C   VARIABLES:                                                          00331600
C                                                                       00331700
C        ITOL = 2 SPECIFIES THAT RTOL IS A SCALAR, ATOL AN ARRAY, AND   00331800
C               THE WEITHTED ERROR IS GIVEN BY:                         00331900
C                         EWT(I) = RTOL + ABS(Y(I)) + ATOL(I)           00332000
C                                                                       00332100
C        RTOL = RELATIVE ERROR IN Y                                     00332200
C                                                                       00332300
C        ATOL = ABSOLUTE ERROR IN Y                                     00332400
C                                                                       00332500
C       ITASK = 1 SPECIFIES THAT THE OUTPUT VALUE OF Y(T) IS            00332600
C               CALCULATED BY OVERSHOOTING AND INTERPOLATING.           00332700
C                                                                       00332800
C      ISTATE = 1 SPECIFIES THAT THIS IS THE FIRST CALL TO LSODA AND    00332900
C               INITIALIZATION IS TO BE DONE.                           00333000
C             = 2 SPECIFIES THAT THIS IS NOT THE FIRST CALL TO LSODA    00333100
C               AND CALCULATIONS ARE TO CONTINUE NORMALLY.              00333200
C               IF ISTATE = 2, THE PARAMETERS THAT ARE ALLOWED TO       00333300
C               BE CHANGED ARE TOUT AND ITASK.                          00333400
C                                                                       00333500
C        IOPT = 0 SPECIFIES THAT NO OPTIONAL INPUTS ARE TO BE USED.     00333600
C                                                                       00333700
C          JT = 2 SPECIFIES THAT AN INTERNALLY SUPPLIED (DIFFERENCE     00333800
C               QUOTIENT) FULL JACOBIAN IS TO USED.                     00333900
C                                                                       00334000
C  NFLAG(INDX)= TRACKS THE NUMBER OF CALLS TO ODE.                      00334100
C                                                                       00334800
C*********************************************************************  00334900
C                                                                       00334910
      EXTERNAL FEX, JDUM, GEX                                           00335000
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00335100
     *          ,NMRCTR                                                 00335200
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7) ,FAO,P,VFLO               00335300
     *        ,IPHASE,MAXCMP,INERT(5)                                   00335400
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                   00335500
     *        ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS   00335600
     *        ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)       00335700
     *        ,RXN(5,14)                                                00335800
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00335900
     *          , ITRS,ITRT,MAXITR,CTOL                                 00336000
      COMMON /INOUT/NW,NR                                               00336100
      COMMON /WORK/SPFRAC(9),STFLO(9,9),RCTVOL(5),                      00336200
     *          CMP(100)                                                00336300
      COMMON /RXSTRM/TOTFLO,RATE,RA                                     00336400
      DOUBLE PRECISION RATE(5),RLS, RLSA,RLSR,FLONRT                    00336500
      DOUBLE PRECISION TOTFLO,ATOL, RWORK, RTOL, T, TOUT, Y, XINCR      00336600
      INTEGER FLGERR,UNTFLG                                             00336700
      CHARACTER RXN                                                     00336900
      DIMENSION RSAV(1), ISAV(1)                                        00337000
      DIMENSION Y(9),NIN(5),NOUT(5),RWORK(215), IWORK(32), ATOL(10)     00337100
      DATA (RWORK(I),I=5,10),(IWORK(J),J=5,10)/6*0.D0,6*0/              00337800
      NFLAG(INDX)=NFLAG(INDX)+1                                         00338501
      NEQ=NRXNTS                                                        00338700
      RWORK(6)=7000.D0                                                  00339100
      LRN=26 + 16*NEQ                                                   00339200
      LRS=28 + 9*NEQ + NEQ**2                                           00339300
      LRW=MAXO(LRN,LRS)                                                 00339400
      LIW=32                                                            00339500
      ITOL = 2                                                          00339600
      RTOL = 1.0D-4                                                     00339700
      DO 10 I=1,10                                                      00339800
10       ATOL(I) = 1.0D-6                                               00339900
      ITASK = 1                                                         00340000
      IOPT = 1                                                          00340200
      JT = 2                                                            00340300
      NG=1                                                              00340400
      TOUT = 0.001D0                                                    00340500
      ISTATE = 1                                                        00340600
      MF=21                                                             00340700
```

```
CU                                                    TOUT{L}                  00340800
      DO 11 I=1,NRXNTS                                                         00341000
 11      Y(I)=RCT(INDX,2,I)*1000.                                              00341100
CU                                                    Y{MOL/MIN}               00341200
      IF(NOPCND(INDX) .EQ. 2) THEN                                             00341300
         NEQ=NEQ+1                                                             00341400
         Y(NEQ)=RCT(INDX,2,13)                                                 00341500
         TEMPK=Y(NEQ)                                                          00341600
         LRW=LRW + 28                                                          00341700
      END IF                                                                   00341800
      VFLO=RCT(INDX,2,10)*1000.*.08206*RCT(INDX,2,13)/P                        00341900
CU    VFLO{L/MIN}                                                             00342000
      T = 0.0D0                                                                00342100
CU    T{L}                                                                    00342200
      DO 12 IOUT = 1,100                                                       00342300
50        CALL LSODAR (FEX,NEQ,Y,T,TOUT,ITOL,RTOL,ATOL,ITASK,ISTATE,           00342400
     1               IOPT,RWORK,LRW,IWORK,LIW,JDUM,JT,                        00342500
     2            GEX, NG, JROOT, INDX)                                       00342600
          IF (ISTATE .LT. 0) GO TO 19                                          00342700
          IF(ISTATE .EQ. 3)  GOTO 133                                          00342800
          IF(ISTATE .EQ. 2) GOTO 14                                            00342900
CU                                                    FA{MOL/MIN}              00343000
          GOTO 50                                                              00343100
CU                                                    FAO{MOL/MIN}             00343200
14        TOUT = TOUT *5.0D0                                                   00343300
12    CONTINUE                                                                 00343400
133   RCTVOL(INDX)=T                                                           00343800
19    CONTINUE                                                                 00343900
      GOTO 20                                                                  00344000
C                                                                              00344100
C   PUT THE FINAL VALUES FROM THE INTEGRATION INTO STRMID                      00344200
C   PUT THE "Y" VALUES (FROM THE INTEGRATION) INTO STRMID                      00344300
C   FROM THE ROUTINE ODE. (Y AND NEQ ARE NOT PASSED TO PFR OR                  00344400
C   ANY OTHER REACTOR.                                                         00344500
C                                                                              00344600
20    DO 21 I=1,100                                                            00344810
         IF(RCT(INDX,I,1) .LE. 0.) GOTO 22                                     00344820
21    CONTINUE                                                                 00344821
22    NN=I                                                                     00344830
      DO 23 I=1,NRXNTS                                                         00344840
         STRMID(NOUT(1),I,1)=RCT(INDX,1,I)                                     00344841
         STRMID(NOUT(1),I,2)=Y(I)/1000.                                        00344842
23       RCT(INDX,NN-1,I)=Y(I)/1000.                                           00344850
      RCT(INDX,NN-1,10)=TOTFLO/1000.                                           00344860
      RCT(INDX,NN-1,11)=X(INDX)                                                00344870
      RCT(INDX,NN-1,12)=RCTVOL(INDX)                                           00344880
      IF(NOPCND(INDX) .EQ. 2) RCT(INDX,NN-1,13)=STRMID(NOUT(1),10,1)           00344890
      RCT(INDX,NN-1,14)=-1./RA                                                 00344891
      DO 16 I=1,5                                                              00345300
         IF(INERT(I) .EQ. 0) GOTO 17                                           00345400
         STRMID(NOUT(1),NRXNTS+I,1)=FLOAT(INERT(I))                            00345500
         STRMID(NOUT(1),NRXNTS+I,2)=FLONRT(I)/1000.                            00345600
16    CONTINUE                                                                 00345700
17    IF(NOPCND(INDX) .EQ. 1) THEN                                             00345800
         STRMID(NOUT(1),10,1)=STRMID(NIN(1),10,1)                              00345900
      ELSE                                                                     00346000
         STRMID(NOUT(1),10,1)=Y(NRXNTS+1)                                      00346100
      END IF                                                                   00346200
      RETURN                                                                   00346300
      END                                                                      00346400
      SUBROUTINE GEX (NEQ,T,Y,NG,GOUT,INDX)                                    00346600
C                                                                              00346610
C**********************************************************************        00346620
C                                                                              00346630
C   SUBROUTINE GEX SUPPLIES THE CONSTRAINT FUNCTION:                           00346640
C                                                                              00346650
C            1.0-X(INDX)-Y(1)/FAO=0.                                           00346660
C                                                                              00346670
C   WHEN THIS EQUATION IS SATISFIED, LSODAR STOPS THE INTEGRATION.             00346680
C   AT THIS POINT, THE CALCULATED CONVERSION IS = THE DESIRED CONVERSION       00346690
C   X(INDX).                                                                   00346691
C**********************************************************************        00346692
```

```
C                                                                       00346693
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7) ,FAO,P,VFLO               00346700
     *         ,IPHASE,MAXCMP,INERT(5)                                  00346800
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                   00346900
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS  00347000
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)      00347100
     *         ,RXN(5,14)                                               00347200
      DOUBLE PRECISION T,Y,GOUT,DFAO,DXINDX,FLONRT                      00347300
      CHARACTER RXN                                                     00347400
      DIMENSION Y(NEQ),GOUT(1)                                          00347500
      DFAO=FAO                                                          00347800
      DXINDX=X(INDX)                                                    00348000
      GOUT(1)=1.DO-Y(1)/DFAO-DXINDX                                     00348100
      RETURN                                                            00348300
      END                                                               00348400
      FUNCTION CALK(A,E,T)                                              00348600
      CALK=A*EXP(-E/1.987/T)                                            00348700
      RETURN                                                            00348800
      END                                                               00348900
      SUBROUTINE FEX (NEQ, T, Y, YDOT,INDX)                             00349000
C                                                                       00349100
C***********************************************************************00349200
C                                                                       00349300
C   VARIABLES:                                                          00349400
C                                                                       00349500
C          C(I) = CONCENTRATION TERM IN RATE EXPRESSION. I=1, NUMBER    00349600
C                 OF COMPONENTS INREACTIONS.                            00349700
C                                                                       00349800
C        CTT(I) = TOTAL CONCENTRATION TERM IN RATE EXPRESSION.          00349900
C                 PRODUCT OF ALL THE CONCENTRATION TERMS. I=1, NUMBER   00350000
C                 OF COMPONENTS IN THE REACTIONS.                       00350100
C                                                                       00350200
C            CT = CONCENTRATION TERM IN THE RATE EXPRESSION.            00350300
C                 THIS TERM INCLUDES ANY EXPONENT THE THE CONCENTRATION 00350400
C                 TERM IS RAISED TO.                                    00350500
C                                                                       00350600
C          NSPC = TOTAL NUMBER OF COMPONENTS IN THE REACTIONS.          00350700
C                                                                       00350800
C       RATE(I) = RATE OF DISAPPEARANCE OF THE KEY COMPONENT OF THE ITH 00350900
C                 REACTION. THE KEY COMPONENT IS THE FIRST COMPONENT    00351000
C                 ENTERED IN A GIVEN REACTION.                          00351100
C                                                                       00351200
C     YDOTI(I) = YDOT INTERNAL. THIS IS USED AS A SORT OF DUMMY         00351300
C                VARIABLE FOR YDOT. SINCE THIS PROGRAM IS NOT           00351400
C                WORKING, THE ONLY IDEA I CAN COME UP WITH IS           00351500
C                THAT I AM DOING TO MANY OPERATIONS ON YDOT.            00351600
C                THEREFORE, I WILL USE THIS INTERNAL VARIABLE           00351700
C                FOR ALL THE CALCULATIONS, THEN SET YDOT=YDOTI          00351800
C                AT THE VERY END OF FEX.                                00351900
C                                                                       00352000
C          INDX = TELLS WHICH UNIT IS BEING CALCULATED.                 00352100
C                                                                       00352200
C***********************************************************************00352300
C                                                                       00352310
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00352400
     *         ,IPHASE,MAXCMP,INERT(5)                                  00352500
      COMMON /LIBDAT/NLIB,NDUM,CPROPS(22,61)                            00352600
      COMMON /TOP/NPM(9,7),NSCM(9,3),NSTRMS,MAXUID,NUID(9),NOPCND(5)    00352700
     *           ,NMRCTR                                                00352800
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                   00352900
     *         ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS  00353000
     *         ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)      00353100
     *         ,RXN(5,14)                                               00353200
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00353300
     *             , ITRS,ITRT,MAXITR,CTOL                              00353400
      COMMON /RXSTRM/TOTFLO,RATE,RA                                     00353500
      INTEGER FLGERR,UNTFLG                                             00353600
      CHARACTER RXN                                                     00353700
      REAL*8 CPROPS                                                     00353800
      DOUBLE PRECISION T, Y, YDOT, C, RATE, CT, CTT                     00353900
     *                ,DEXP,DVFLO,DFAO,COEFP,COEFR                      00354000
     *                ,YDOTI,TOTFLO,FLONRT,CP,SUMCCP,SUMHR              00354100
```

```
     *                ,DTEMPK                                                  00354200
      DIMENSION Y(NEQ), YDOT(NEQ),C(8), RATE(5),YDOTI(8)                      00354400
      IF(T .EQ. O.ODO) NCOUNT=3                                               00355210
      DO 9 I=1,NRXNS                                                          00355500
9        RATE(I)=O.ODO                                                        00355600
      SUMCCP=O.DO                                                             00355700
C                                                                             00355800
C   FIND TOTAL MOLAR FLOW RATE                                                00355900
C                                                                             00356000
      TOTFLO=O.DO                                                             00356200
      XA=1.-Y(1)/FAO                                                          00356300
      IF(XA .LE. X(INDX) .AND. NOPCND(INDX) .EQ. 2) TEMPK=Y(NEQ)              00356500
      IF(NOPCND(INDX) .EQ. 1) TEMPK=RCT(INDX,2,13)                            00356600
      DTEMPK=TEMPK                                                            00356700
CU                                          TOTFLO{MOL/MIN}                 00356900
      DO 15 I=1,NRXNTS                                                        00357000
15       TOTFLO=TOTFLO+Y(I)                                                   00357100
      TOTFLO=TOTFLO+FLONRT(6)                                                 00357200
CU                                          FLONRT{MOL/MIN}                 00357300
      DVFLO=TOTFLO*8.206D-2*DTEMPK/P                                          00357700
CU                                          TEMPK{K}                        00358000
C                                                                             00358100
C   SET DOUBLE PRECISION VALUES AS NEEDED                                     00358200
C                                                                             00358300
      DFAO=FAO                                                                00358400
      VFLO=DVFLO                                                              00358500
      JJ=NEQ                                                                  00358600
      XA=1.-Y(1)/DFAO                                                         00358800
      DO 10 I=1,NRXNTS                                                        00359200
         C(I)=Y(I)/DVFLO                                                      00359400
         IF(NOPCND(INDX) .EQ. 2) THEN                                         00360100
            CP=O.DO                                                           00360200
            DO 17 II=1,7                                                      00360300
17          CP=CP + CPROPS(II+15,NCOEF(I,1))*(Y(NEQ)**(II-1))                 00360700
            SUMCCP=SUMCCP + C(I)*CP                                           00361000
         END IF                                                               00361100
10    CONTINUE                                                                00361200
      DO 35 I=1,5                                                             00361800
         IF(INERT(I) .EQ. O .OR. NOPCND(INDX) .EQ. 1) GOTO 37                 00361900
         CP=O.DO                                                              00362000
            DO 36 II=1,7                                                      00362100
36          CP=CP + CPROPS(II+15,INERT(I))*(Y(NEQ)**(II-1))                   00362200
         C(I)=FLONRT(I)/DVFLO                                                 00362400
         SUMCCP=SUMCCP + C(I)*CP                                              00362500
35    CONTINUE                                                                00362600
37    SUMCCP=SUMCCP*DVFLO                                                     00362900
C                                                                             00363100
C   THE SUMCCP TERM REPRESENTS : VOL FLO RATE*SUM(C(I)*CP(I))                 00363200
C                                                                             00363300
      DO 11 I=1,NRXNS                                                         00363500
         CT=1.ODO                                                             00363700
         J=1                                                                  00363800
33       IF(J .LE. NPOSPC(6,I)) THEN                                          00363900
            DEXP=EXP(NPOSPC(J,I),I)                                           00364100
            CT=CT*C(NPOSPC(J,I))**DEXP                                        00364300
            J=J+1                                                             00364700
            GOTO 33                                                           00364800
         END IF                                                               00364900
         IF(NOPCND(INDX) .EQ. 2) THEN                                         00365100
            AK=A(I)                                                           00365200
            ACTEN=E(I)                                                        00365300
            DRK=CALK(AK,ACTEN,TEMPK)                                          00365400
            GOTO 20                                                           00365800
         END IF                                                               00365900
         DRK=RK(I,1)                                                          00366000
20       CT=CT*DRK                                                            00366200
         RATE(I)=CT                                                           00366300
         IF( I .EQ. 1) THEN                                                   00366400
         END IF                                                               00366800
CU    RATE{MOL-L-MIN}                                                         00366900
         IF(IDIR(I) .EQ. O) GOTO 11                                           00367000
         NSPC=NPOSPC(6,I)+NPOSPC(7,I)                                         00367100
```

```
          CT=1.0D0                                                          00367200
34        IF(J .LE. NSPC) THEN                                              00367300
             DEXP=EXP(NPOSPC(J,I),I)                                        00367400
             CT=CT*C(NPOSPC(J,I))**DEXP                                     00367700
             J=J+1                                                          00367800
             GOTO 34                                                        00367900
          END IF                                                            00368000
          CT=CT*RK(I,1)/RK(I,2)                                             00368200
          RATE(I)=RATE(I)-CT                                                00368300
11      CONTINUE                                                            00368500
C                                                                           00368600
C     NOW HAVE ALL THE RATE EXPRESSIONS.                                    00368700
C                                                                           00368800
C                                                                           00368900
C     NOW CREATE THE DIFFENTIAL EXPRESSIONS (DF(I)/DV   I=1,# COMPONENTS).00369000
C     DO THE NONKEY COMPONENT FIRST.                                        00369300
C                                                                           00369400
          DO 12 I=1,NRXNTS                                                  00369500
12           YDOTI(I)=0.0D0                                                 00369600
          I=1                                                               00369700
32        IF(NCOEF(I,1) .NE. 0) THEN                                        00369900
CU      YDOT{MOL-L-MIN}                                                     00370000
          DO 13 J=2,6                                                       00370100
             IF(NCOEF(I,J) .NE. 0) THEN                                     00370500
                IF(KEYPOS(J-1) .EQ. I) GOTO 14                              00370600
                COEFP=FLOAT(NCOEF(I,J))                                     00370700
                COEFR=FLOAT(IABS(NCOEF(KEYPOS(J-1),J)))                     00370800
                YDOTI(I)=YDOTI(I)+COEFP/COEFR*RATE(J-1)                     00370900
                GOTO 13                                                     00371000
14              YDOTI(I)=YDOTI(I)-RATE(J-1)                                 00371100
             END IF                                                         00371300
13        CONTINUE                                                          00371400
           I=I+1                                                            00371500
           IF(I .EQ. 10) GOTO 16                                            00371600
        GOTO 32                                                             00371700
        END IF                                                              00371800
CU      VFLO{L/MIN}                                                         00371900
16      I=1                                                                 00372300
30      IF(NCOEF(I,1) .NE. 0) THEN                                          00372400
           YDOT(I)=YDOTI(I)                                                 00372500
           I=I+1                                                            00372700
           GOTO 30                                                          00372800
        END IF                                                              00372900
C                                                                           00373000
C     SUMHR=SUM(DELTAH(I) * RATE(I)) TERM IN THE ENERGY BALANCE             00373100
C                                                                           00373200
        RA=YDOT(1)                                                          00373300
        IF(NOPCND(INDX) .EQ. 1) GOTO 19                                     00373400
        SUMHR=0.0D0                                                         00373500
        DO 18 I=1,NRXNS                                                     00373600
18         SUMHR=SUMHR + RATE(I)*DBLE(DELH(I))                              00373700
        YDOT(NEQ)=-SUMHR/SUMCCP                                             00373900
19      IF(XA .GT. X(INDX)) GOTO 21                                         00374110
        IF(DABS(Y(1)-RCT(INDX,NCOUNT-1,1)*1000.)/Y(1) .LE. .01) GOTO 21     00374400
        DO 38 IU=1,NRXNTS                                                   00374500
38         RCT(INDX,NCOUNT,IU)=Y(IU)/1000.0D0                               00374600
        RCT(INDX,NCOUNT,10)=TOTFLO/1000.0D0                                 00374700
        RCT(INDX,NCOUNT,11)=1.-Y(1)/FA0                                     00374800
        RCT(INDX,NCOUNT,12)= T                                              00374900
        RCT(INDX,NCOUNT,14)=-1./YDOT(1)                                     00374910
        IF(NOPCND(INDX) .EQ. 2) RCT(INDX,NCOUNT,13)=Y(NEQ)                  00375000
        NCOUNT=NCOUNT + 1                                                   00375100
21      RETURN                                                              00375300
        END                                                                 00375400
        SUBROUTINE LSODAR (F, NEQ, Y, T, TOUT, ITOL, RTOL, ATOL, ITASK,     00375600
       1             ISTATE, IOPT, RWORK, LRW, IWORK, LIW, JAC, JT,       00375700
       2             G, NG, JROOT,INDX)                                   00375800
        EXTERNAL F, JAC, G                                                  00375900
        INTEGER NEQ, ITOL, ITASK, ISTATE, IOPT, LRW, IWORK, LIW, JT,        00376000
       1   NG, JROOT                                                        00376100
        DOUBLE PRECISION Y, T, TOUT, RTOL, ATOL, RWORK                      00376200
        DIMENSION  Y(1), RTOL(1), ATOL(1), RWORK(LRW), IWORK(LIW),          00376300
```

```
CN                                                                      00376400
     1   JROOT(NG)                                                      00376500
C-----------------------------------------------------------------------00376600
C THIS IS THE MAY 7, 1982 VERSION OF                                    00376700
C LSODAR.. LIVERMORE SOLVER FOR ORDINARY DIFFERENTIAL EQUATIONS, WITH   00376800
C          AUTOMATIC METHOD SWITCHING FOR STIFF AND NONSTIFF PROBLEMS,  00376900
C          AND WITH ROOT-FINDING.                                       00377000
C                                                                       00377100
C THIS VERSION IS IN DOUBLE PRECISION.                                  00377200
C                                                                       00377300
C LSODAR SOLVES THE INITIAL VALUE PROBLEM FOR STIFF OR NONSTIFF         00377400
C SYSTEMS OF FIRST ORDER ODE-S,                                         00377500
C     DY/DT = F(T,Y) ,  OR, IN COMPONENT FORM,                          00377600
C     DY(I)/DT = F(I) = F(I,T,Y(1),Y(2),...,Y(NEQ)) (I = 1,...,NEQ).    00377700
C AT THE SAME TIME, IT LOCATES THE ROOTS OF ANY OF A SET OF FUNCTIONS   00377800
C     G(I) = G(I,T,Y(1),...,Y(NEQ))  (I = 1,...,NG).                    00377900
C                                                                       00378000
C THIS A VARIANT VERSION OF THE LSODE PACKAGE.  IT DIFFERS FROM LSODE   00378100
C IN TWO WAYS..                                                         00378200
C (A) IT SWITCHES AUTOMATICALLY BETWEEN STIFF AND NONSTIFF METHODS.     00378300
C THIS MEANS THAT THE USER DOES NOT HAVE TO DETERMINE WHETHER THE       00378400
C PROBLEM IS STIFF OR NOT, AND THE SOLVER WILL AUTOMATICALLY CHOOSE THE 00378500
C APPROPRIATE METHOD.  IT ALWAYS STARTS WITH THE NONSTIFF METHOD.       00378600
C (B) IT FINDS THE ROOT OF AT LEAST ONE OF A SET OF CONSTRAINT          00378700
C FUNCTIONS G(I) OF THE INDEPENDENT AND DEPENDENT VARIABLES.            00378800
C IT FINDS ONLY THOSE ROOTS FOR WHICH SOME G(I), AS A FUNCTION          00378900
C OF T, CHANGES SIGN IN THE INTERVAL OF INTEGRATION.                    00379000
C IT THEN RETURNS THE SOLUTION AT THE ROOT, IF THAT OCCURS              00379100
C SOONER THAN THE SPECIFIED STOP CONDITION, AND OTHERWISE RETURNS       00379200
C THE SOLUTION ACCORDING THE SPECIFIED STOP CONDITION.                  00379300
C                                                                       00379400
C AUTHORS..                                                             00379500
C                       LINDA R. PETZOLD                                00379600
C                       APPLIED MATHEMATICS DIVISION 8331               00379700
C                       SANDIA NATIONAL LABORATORIES                    00379800
C                       LIVERMORE, CA 94550                             00379900
C AND                                                                   00380000
C                       ALAN C. HINDMARSH,                              00380100
C                       MATHEMATICS AND STATISTICS DIVISION, L-316      00380200
C                       LAWRENCE LIVERMORE NATIONAL LABORATORY          00380300
C                       LIVERMORE, CA 94550.                            00380400
C                                                                       00380500
C REFERENCES..                                                          00380600
C 1.  ALAN C. HINDMARSH,  LSODE AND LSODI, TWO NEW INITIAL VALUE        00380700
C     ORDINARY DIFFERENTIAL EQUATION SOLVERS,                           00380800
C     ACM-SIGNUM NEWSLETTER, VOL. 15, NO. 4 (1980), PP. 10-11.          00380900
C 2.  LINDA R. PETZOLD, AUTOMATIC SELECTION OF METHODS FOR SOLVING      00381000
C     STIFF AND NONSTIFF SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS,    00381100
C     SANDIA NATIONAL LABORATORIES REPORT SAND80-8230, SEPTEMBER 1980.  00381200
C 3.  KATHIE L. HIEBERT AND LAWRENCE F. SHAMPINE, IMPLICITLY DEFINED   00381300
C     OUTPUT POINTS FOR SOLUTIONS OF ODE-S, SANDIA REPORT SAND80-0180,  00381400
C     FEBRUARY, 1980.                                                   00381500
C-----------------------------------------------------------------------00381600
C SUMMARY OF USAGE.                                                     00381700
C                                                                       00381800
C COMMUNICATION BETWEEN THE USER AND THE LSODAR PACKAGE, FOR NORMAL     00381900
C SITUATIONS, IS SUMMARIZED HERE.  THIS SUMMARY DESCRIBES ONLY A SUBSET 00382000
C OF THE FULL SET OF OPTIONS AVAILABLE.  SEE THE FULL DESCRIPTION FOR   00382100
C DETAILS, INCLUDING ALTERNATIVE TREATMENT OF THE JACOBIAN MATRIX,      00382200
C OPTIONAL INPUTS AND OUTPUTS, NONSTANDARD OPTIONS, AND                 00382300
C INSTRUCTIONS FOR SPECIAL SITUATIONS.  SEE ALSO THE EXAMPLE            00382400
C PROBLEM (WITH PROGRAM AND OUTPUT) FOLLOWING THIS SUMMARY.             00382500
C                                                                       00382600
C A. FIRST PROVIDE A SUBROUTINE OF THE FORM..                           00382700
C               SUBROUTINE F (NEQ, T, Y, YDOT)                          00382800
C               DIMENSION Y(NEQ), YDOT(NEQ)                             00382900
C WHICH SUPPLIES THE VECTOR FUNCTION F BY LOADING YDOT(I) WITH F(I).    00383000
C                                                                       00383100
C B. PROVIDE A SUBROUTINE OF THE FORM..                                 00383200
C               SUBROUTINE G (NEQ, T, Y, NG, GOUT)                      00383300
C               DIMENSION Y(NEQ), GOUT(NG)                              00383400
C WHICH SUPPLIES THE VECTOR FUNCTION G BY LOADING GOUT(I) WITH          00383500
```

```
C G(I), THE I-TH CONSTRAINT FUNCTION WHOSE ROOT IS SOUGHT.                00383600
C                                                                        00383700
C C. WRITE A MAIN PROGRAM WHICH CALLS SUBROUTINE LSODAR ONCE FOR         00383800
C EACH POINT AT WHICH ANSWERS ARE DESIRED.  THIS SHOULD ALSO PROVIDE     00383900
C FOR POSSIBLE USE OF LOGICAL UNIT 6 FOR OUTPUT OF ERROR MESSAGES BY     00384000
C LSODAR.  ON THE FIRST CALL TO LSODAR, SUPPLY ARGUMENTS AS FOLLOWS..    00384100
C F      = NAME OF SUBROUTINE FOR RIGHT-HAND SIDE VECTOR F.              00384200
C          THIS NAME MUST BE DECLARED EXTERNAL IN CALLING PROGRAM.       00384300
C NEQ    = NUMBER OF FIRST ORDER ODE-S.                                  00384400
C Y      = ARRAY OF INITIAL VALUES, OF LENGTH NEQ.                       00384500
C T      = THE INITIAL VALUE OF THE INDEPENDENT VARIABLE.                00384600
C TOUT   = FIRST POINT WHERE OUTPUT IS DESIRED (.NE. T).                 00384700
C ITOL   = 1 OR 2 ACCORDING AS ATOL (BELOW) IS A SCALAR OR ARRAY.        00384800
C RTOL   = RELATIVE TOLERANCE PARAMETER (SCALAR).                        00384900
C ATOL   = ABSOLUTE TOLERANCE PARAMETER (SCALAR OR ARRAY).               00385000
C          THE ESTIMATED LOCAL ERROR IN Y(I) WILL BE CONTROLLED SO AS    00385100
C          TO BE LESS THAN                                               00385200
C             EWT(I) = RTOL*ABS(Y(I)) + ATOL     IF ITOL = 1, OR        00385300
C             EWT(I) = RTOL*ABS(Y(I)) + ATOL(I)  IF ITOL = 2.           00385400
C          THUS THE LOCAL ERROR TEST PASSES IF, IN EACH COMPONENT,       00385500
C          EITHER THE ABSOLUTE ERROR IS LESS THAN ATOL (OR ATOL(I)),     00385600
C          OR THE RELATIVE ERROR IS LESS THAN RTOL.                      00385700
C          USE RTOL = 0.0 FOR PURE ABSOLUTE ERROR CONTROL, AND           00385800
C          USE ATOL = 0.0 (OR ATOL(I) = 0.0) FOR PURE RELATIVE ERROR     00385900
C          CONTROL.  CAUTION.. ACTUAL (GLOBAL) ERRORS MAY EXCEED THESE   00386000
C          LOCAL TOLERANCES, SO CHOOSE THEM CONSERVATIVELY.              00386100
C ITASK  = 1 FOR NORMAL COMPUTATION OF OUTPUT VALUES OF Y AT T = TOUT.   00386200
C ISTATE = INTEGER FLAG (INPUT AND OUTPUT).  SET ISTATE = 1.             00386300
C IOPT   = 0 TO INDICATE NO OPTIONAL INPUTS USED.                        00386400
C RWORK  = REAL WORK ARRAY OF LENGTH AT LEAST..                          00386500
C             22 + NEQ * MAX(16, NEQ + 9) + 3*NG.                        00386600
C          SEE ALSO PARAGRAPH F BELOW.                                   00386700
C LRW    = DECLARED LENGTH OF RWORK (IN USER-S DIMENSION).               00386800
C IWORK  = INTEGER WORK ARRAY OF LENGTH AT LEAST  20 + NEQ.              00386900
C LIW    = DECLARED LENGTH OF IWORK (IN USER-S DIMENSION).               00387000
C JAC    = NAME OF SUBROUTINE FOR JACOBIAN MATRIX.                       00387100
C          USE A DUMMY NAME.  SEE ALSO PARAGRAPH F BELOW.                00387200
C JT     = JACOBIAN TYPE INDICATOR.  SET JT = 2.                         00387300
C          SEE ALSO PARAGRAPH F BELOW.                                   00387400
C G      = NAME OF SUBROUTINE FOR CONSTRAINT FUNCTIONS, WHOSE            00387500
C          ROOTS ARE DESIRED DURING THE INTEGRATION.                     00387600
C          THIS NAME MUST BE DECLARED EXTERNAL IN CALLING PROGRAM.       00387700
C NG     = NUMBER OF CONSTRAINT FUNCTIONS G(I).  IF THERE ARE NONE,      00387800
C          SET NG = 0, AND PASS A DUMMY NAME FOR G.                      00387900
C JROOT  = INTEGER ARRAY OF LENGTH NG FOR OUTPUT OF ROOT INFORMATION.    00388000
C          SEE NEXT PARAGRAPH.                                           00388100
C NOTE THAT THE MAIN PROGRAM MUST DECLARE ARRAYS Y, RWORK, IWORK,        00388200
C JROOT, AND POSSIBLY ATOL.                                              00388300
C                                                                        00388400
C D. THE OUTPUT FROM THE FIRST CALL (OR ANY CALL) IS..                   00388500
C      Y = ARRAY OF COMPUTED VALUES OF Y(T) VECTOR.                      00388600
C      T = CORRESPONDING VALUE OF INDEPENDENT VARIABLE.  THIS IS         00388700
C          TOUT IF ISTATE = 2, OR THE ROOT LOCATION IF ISTATE = 3,       00388800
C          OR THE FARTHEST POINT REACHED IF LSODAR WAS UNSUCCESSFUL.     00388900
C ISTATE = 2 OR 3  IF LSODAR WAS SUCCESSFUL, NEGATIVE OTHERWISE.         00389000
C           2 MEANS NO ROOT WAS FOUND, AND TOUT WAS REACHED AS DESIRED.  00389100
C           3 MEANS A ROOT WAS FOUND PRIOR TO REACHING TOUT.             00389200
C          -1 MEANS EXCESS WORK DONE ON THIS CALL (PERHAPS WRONG JT).    00389300
C          -2 MEANS EXCESS ACCURACY REQUESTED (TOLERANCES TOO SMALL).    00389400
C          -3 MEANS ILLEGAL INPUT DETECTED (SEE PRINTED MESSAGE).        00389500
C          -4 MEANS REPEATED ERROR TEST FAILURES (CHECK ALL INPUTS).     00389600
C          -5 MEANS REPEATED CONVERGENCE FAILURES (PERHAPS BAD JACOBIAN  00389700
C             SUPPLIED OR WRONG CHOICE OF JT OR TOLERANCES).             00389800
C          -6 MEANS ERROR WEIGHT BECAME ZERO DURING PROBLEM. (SOLUTION   00389900
C             COMPONENT I VANISHED, AND ATOL OR ATOL(I) = 0.)            00390000
C          -7 MEANS WORK SPACE INSUFFICIENT TO FINISH (SEE MESSAGES).    00390100
C JROOT  = ARRAY SHOWING ROOTS FOUND IF ISTATE = 3 ON RETURN.            00390200
C          JROOT(I) = 1 IF G(I) HAS A ROOT AT T, OR 0 OTHERWISE.         00390300
C                                                                        00390400
C E. TO CONTINUE THE INTEGRATION AFTER A SUCCESSFUL RETURN, PROCEED      00390500
C AS FOLLOWS..                                                           00390600
C  (A) IF ISTATE = 2 ON RETURN, RESET TOUT AND CALL LSODAR AGAIN.        00390700
```

```
C  (B) IF ISTATE = 3 ON RETURN, RESET ISTATE TO 2 AND CALL LSODAR AGAIN.00390800
C IN EITHER CASE, NO OTHER PARAMETERS NEED BE RESET.                    00390900
C                                                                       00391000
C F. NOTE.. IF AND WHEN LSODAR REGARDS THE PROBLEM AS STIFF, AND        00391100
C SWITCHES METHODS ACCORDINGLY, IT MUST MAKE USE OF THE NEQ BY NEQ      00391200
C JACOBIAN MATRIX, J = DF/DY.  FOR THE SAKE OF SIMPLICITY, THE          00391300
C INPUTS TO LSODAR RECOMMENDED IN PARAGRAPH C ABOVE CAUSE LSODAR TO     00391400
C TREAT J AS A FULL MATRIX, AND TO APPROXIMATE IT INTERNALLY BY         00391500
C DIFFERENCE QUOTIENTS.  ALTERNATIVELY, J CAN BE TREATED AS A BAND      00391600
C MATRIX (WITH GREAT POTENTIAL REDUCTION IN THE SIZE OF THE RWORK       00391700
C ARRAY).  ALSO, IN EITHER THE FULL OR BANDED CASE, THE USER CAN SUPPLY 00391800
C J IN CLOSED FORM, WITH A ROUTINE WHOSE NAME IS PASSED AS THE JAC      00391900
C ARGUMENT.  THESE ALTERNATIVES ARE DESCRIBED IN THE PARAGRAPHS ON      00392000
C RWORK, JAC, AND JT IN THE FULL DESCRIPTION OF THE CALL SEQUENCE BELOW.00392100
C                                                                       00392200
C-----------------------------------------------------------------------00392300
C FULL DESCRIPTION OF USER INTERFACE TO LSODAR.                         00402900
C                                                                       00403000
C THE USER INTERFACE TO LSODAR CONSISTS OF THE FOLLOWING PARTS.         00403100
C                                                                       00403200
C I.   THE CALL SEQUENCE TO SUBROUTINE LSODAR, WHICH IS A DRIVER        00403300
C      ROUTINE FOR THE SOLVER.  THIS INCLUDES DESCRIPTIONS OF BOTH      00403400
C      THE CALL SEQUENCE ARGUMENTS AND OF USER-SUPPLIED ROUTINES.       00403500
C      FOLLOWING THESE DESCRIPTIONS IS A DESCRIPTION OF                 00403600
C      OPTIONAL INPUTS AVAILABLE THROUGH THE CALL SEQUENCE, AND THEN    00403700
C      A DESCRIPTION OF OPTIONAL OUTPUTS (IN THE WORK ARRAYS).          00403800
C                                                                       00403900
C II.  DESCRIPTIONS OF OTHER ROUTINES IN THE LSODAR PACKAGE THAT MAY BE 00404000
C      (OPTIONALLY) CALLED BY THE USER.  THESE PROVIDE THE ABILITY TO   00404100
C      ALTER ERROR MESSAGE HANDLING, SAVE AND RESTORE THE INTERNAL      00404200
C      COMMON, AND OBTAIN SPECIFIED DERIVATIVES OF THE SOLUTION Y(T).   00404300
C                                                                       00404400
C III. DESCRIPTIONS OF COMMON BLOCKS TO BE DECLARED IN OVERLAY          00404500
C      OR SIMILAR ENVIRONMENTS, OR TO BE SAVED WHEN DOING AN INTERRUPT  00404600
C      OF THE PROBLEM AND CONTINUED SOLUTION LATER.                     00404700
C                                                                       00404800
C IV.  DESCRIPTION OF TWO SUBROUTINES IN THE LSODAR PACKAGE, EITHER OF  00404900
C      WHICH THE USER MAY REPLACE WITH HIS OWN VERSION, IF DESIRED.     00405000
C      THESE RELATE TO THE MEASUREMENT OF ERRORS.                       00405100
C                                                                       00405200
C-----------------------------------------------------------------------00405300
C PART I.  CALL SEQUENCE.                                               00405400
C                                                                       00405500
C THE CALL SEQUENCE PARAMETERS USED FOR INPUT ONLY ARE                  00405600
C     F, NEQ, TOUT, ITOL, RTOL, ATOL, ITASK, IOPT, LRW, LIW, JAC,       00405700
C     JT, G, AND NG,                                                    00405800
C THAT USED ONLY FOR OUTPUT IS  JROOT,                                  00405900
C AND THOSE USED FOR BOTH INPUT AND OUTPUT ARE                          00406000
C     Y, T, ISTATE.                                                     00406100
C THE WORK ARRAYS RWORK AND IWORK ARE ALSO USED FOR CONDITIONAL AND     00406200
C OPTIONAL INPUTS AND OPTIONAL OUTPUTS.  (THE TERM OUTPUT HERE REFERS   00406300
C TO THE RETURN FROM SUBROUTINE LSODAR TO THE USER-S CALLING PROGRAM.)  00406400
C                                                                       00406500
C THE LEGALITY OF INPUT PARAMETERS WILL BE THOROUGHLY CHECKED ON THE    00406600
C INITIAL CALL FOR THE PROBLEM, BUT NOT CHECKED THEREAFTER UNLESS A     00406700
C CHANGE IN INPUT PARAMETERS IS FLAGGED BY ISTATE = 3 ON INPUT.         00406800
C                                                                       00406900
C THE DESCRIPTIONS OF THE CALL ARGUMENTS ARE AS FOLLOWS.                00407000
C                                                                       00407100
C F       = THE NAME OF THE USER-SUPPLIED SUBROUTINE DEFINING THE       00407200
C           ODE SYSTEM.  THE SYSTEM MUST BE PUT IN THE FIRST-ORDER      00407300
C           FORM DY/DT = F(T,Y), WHERE F IS A VECTOR-VALUED FUNCTION    00407400
C           OF THE SCALAR T AND THE VECTOR Y.  SUBROUTINE F IS TO       00407500
C           COMPUTE THE FUNCTION F.  IT IS TO HAVE THE FORM             00407600
C                SUBROUTINE F (NEQ, T, Y, YDOT)                         00407700
C                DIMENSION Y(1), YDOT(1)                                00407800
C           WHERE NEQ, T, AND Y ARE INPUT, AND THE ARRAY YDOT = F(T,Y)  00407900
C           IS OUTPUT.  Y AND YDOT ARE ARRAYS OF LENGTH NEQ.            00408000
C           (IN THE DIMENSION STATEMENT ABOVE, 1 IS A DUMMY             00408100
C           DIMENSION.. IT CAN BE REPLACED BY ANY VALUE.)               00408200
C           SUBROUTINE F SHOULD NOT ALTER Y(1),...,Y(NEQ).              00408300
C           F MUST BE DECLARED EXTERNAL IN THE CALLING PROGRAM.         00408400
```

```
C                                                                       00408500
C          SUBROUTINE F MAY ACCESS USER-DEFINED QUANTITIES IN           00408600
C          NEQ(2),... AND Y(NEQ(1)+1),... IF NEQ IS AN ARRAY            00408700
C          (DIMENSIONED IN F) AND Y HAS LENGTH EXCEEDING NEQ(1).        00408800
C          SEE THE DESCRIPTIONS OF NEQ AND Y BELOW.                     00408900
C                                                                       00409000
C NEQ    = THE SIZE OF THE ODE SYSTEM (NUMBER OF FIRST ORDER            00409100
C          ORDINARY DIFFERENTIAL EQUATIONS).  USED ONLY FOR INPUT.      00409200
C          NEQ MAY BE DECREASED, BUT NOT INCREASED, DURING THE PROBLEM. 00409300
C          IF NEQ IS DECREASED (WITH ISTATE = 3 ON INPUT), THE          00409400
C          REMAINING COMPONENTS OF Y SHOULD BE LEFT UNDISTURBED, IF     00409500
C          THESE ARE TO BE ACCESSED IN F AND/OR JAC.                    00409600
C                                                                       00409700
C          NORMALLY, NEQ IS A SCALAR, AND IT IS GENERALLY REFERRED TO   00409800
C          AS A SCALAR IN THIS USER INTERFACE DESCRIPTION.  HOWEVER,    00409900
C          NEQ MAY BE AN ARRAY, WITH NEQ(1) SET TO THE SYSTEM SIZE.     00410000
C          (THE LSODAR PACKAGE ACCESSES ONLY NEQ(1).)  IN EITHER CASE,  00410100
C          THIS PARAMETER IS PASSED AS THE NEQ ARGUMENT IN ALL CALLS    00410200
C          TO F, JAC, AND G.  HENCE, IF IT IS AN ARRAY, LOCATIONS       00410300
C          NEQ(2),... MAY BE USED TO STORE OTHER INTEGER DATA AND PASS  00410400
C          IT TO F, JAC, AND G.  EACH SUCH SUBROUTINE MUST INCLUDE      00410500
C          NEQ IN A DIMENSION STATEMENT IN THAT CASE.                   00410600
C                                                                       00410700
C Y      = A REAL ARRAY FOR THE VECTOR OF DEPENDENT VARIABLES, OF       00410800
C          LENGTH NEQ OR MORE.  USED FOR BOTH INPUT AND OUTPUT ON THE   00410900
C          FIRST CALL (ISTATE = 1), AND ONLY FOR OUTPUT ON OTHER CALLS. 00411000
C          ON THE FIRST CALL, Y MUST CONTAIN THE VECTOR OF INITIAL      00411100
C          VALUES.  ON OUTPUT, Y CONTAINS THE COMPUTED SOLUTION VECTOR, 00411200
C          EVALUATED AT T.  IF DESIRED, THE Y ARRAY MAY BE USED         00411300
C          FOR OTHER PURPOSES BETWEEN CALLS TO THE SOLVER.              00411400
C                                                                       00411500
C          THIS ARRAY IS PASSED AS THE Y ARGUMENT IN ALL CALLS TO F,    00411600
C          JAC, AND G.  HENCE ITS LENGTH MAY EXCEED NEQ, AND LOCATIONS  00411700
C          Y(NEQ+1),... MAY BE USED TO STORE OTHER REAL DATA AND        00411800
C          PASS IT TO F, JAC, AND G.  (THE LSODAR PACKAGE ACCESSES ONLY 00411900
C          Y(1),...,Y(NEQ).)                                            00412000
C                                                                       00412100
C T      = THE INDEPENDENT VARIABLE.  ON INPUT, T IS USED ONLY ON THE   00412200
C          FIRST CALL, AS THE INITIAL POINT OF THE INTEGRATION.         00412300
C          ON OUTPUT, AFTER EACH CALL, T IS THE VALUE AT WHICH A        00412400
C          COMPUTED SOLUTION Y IS EVALUATED (USUALLY THE SAME AS TOUT). 00412500
C          IF A ROOT WAS FOUND, T IS THE COMPUTED LOCATION OF THE       00412600
C          ROOT REACHED FIRST, ON OUTPUT.                               00412700
C          ON AN ERROR RETURN, T IS THE FARTHEST POINT REACHED.         00412800
C                                                                       00412900
C TOUT   = THE NEXT VALUE OF T AT WHICH A COMPUTED SOLUTION IS DESIRED. 00413000
C          USED ONLY FOR INPUT.                                         00413100
C                                                                       00413200
C          WHEN STARTING THE PROBLEM (ISTATE = 1), TOUT MAY BE EQUAL    00413300
C          TO T FOR ONE CALL, THEN SHOULD .NE. T FOR THE NEXT CALL.     00413400
C          FOR THE INITIAL T, AN INPUT VALUE OF TOUT .NE. T IS USED     00413500
C          IN ORDER TO DETERMINE THE DIRECTION OF THE INTEGRATION       00413600
C          (I.E. THE ALGEBRAIC SIGN OF THE STEP SIZES) AND THE ROUGH    00413700
C          SCALE OF THE PROBLEM.  INTEGRATION IN EITHER DIRECTION       00413800
C          (FORWARD OR BACKWARD IN T) IS PERMITTED.                     00413900
C                                                                       00414000
C          IF ITASK = 2 OR 5 (ONE-STEP MODES), TOUT IS IGNORED AFTER    00414100
C          THE FIRST CALL (I.E. THE FIRST CALL WITH TOUT .NE. T).       00414200
C          OTHERWISE, TOUT IS REQUIRED ON EVERY CALL.                   00414300
C                                                                       00414400
C          IF ITASK = 1, 3, OR 4, THE VALUES OF TOUT NEED NOT BE        00414500
C          MONOTONE, BUT A VALUE OF TOUT WHICH BACKS UP IS LIMITED      00414600
C          TO THE CURRENT INTERNAL T INTERVAL, WHOSE ENDPOINTS ARE      00414700
C          TCUR - HU AND TCUR (SEE OPTIONAL OUTPUTS, BELOW, FOR         00414800
C          TCUR AND HU).                                                00414900
C                                                                       00415000
C ITOL   = AN INDICATOR FOR THE TYPE OF ERROR CONTROL.  SEE             00415100
C          DESCRIPTION BELOW UNDER ATOL.  USED ONLY FOR INPUT.          00415200
C                                                                       00415300
C RTOL   = A RELATIVE ERROR TOLERANCE PARAMETER, EITHER A SCALAR OR     00415400
C          AN ARRAY OF LENGTH NEQ.  SEE DESCRIPTION BELOW UNDER ATOL.   00415500
C          INPUT ONLY.                                                  00415600
```

```
C                                                                       00415700
C ATOL    = AN ABSOLUTE ERROR TOLERANCE PARAMETER, EITHER A SCALAR OR   00415800
C           AN ARRAY OF LENGTH NEQ.  INPUT ONLY.                        00415900
C                                                                       00416000
C              THE INPUT PARAMETERS ITOL, RTOL, AND ATOL DETERMINE      00416100
C           THE ERROR CONTROL PERFORMED BY THE SOLVER.  THE SOLVER WILL  00416200
C           CONTROL THE VECTOR E = (E(I)) OF ESTIMATED LOCAL ERRORS      00416300
C           IN Y, ACCORDING TO AN INEQUALITY OF THE FORM                 00416400
C                       MAX-NORM OF ( E(I)/EWT(I) )   .LE.   1,         00416500
C           WHERE EWT = (EWT(I)) IS A VECTOR OF POSITIVE ERROR WEIGHTS.  00416600
C           THE VALUES OF RTOL AND ATOL SHOULD ALL BE NON-NEGATIVE.      00416700
C           THE FOLLOWING TABLE GIVES THE TYPES (SCALAR/ARRAY) OF        00416800
C           RTOL AND ATOL, AND THE CORRESPONDING FORM OF EWT(I).         00416900
C                                                                       00417000
C              ITOL    RTOL       ATOL          EWT(I)                  00417100
C               1     SCALAR     SCALAR     RTOL*ABS(Y(I)) + ATOL       00417200
C               2     SCALAR     ARRAY      RTOL*ABS(Y(I)) + ATOL(I)    00417300
C               3     ARRAY      SCALAR     RTOL(I)*ABS(Y(I)) + ATOL    00417400
C               4     ARRAY      ARRAY      RTOL(I)*ABS(Y(I)) + ATOL(I) 00417500
C                                                                       00417600
C           WHEN EITHER OF THESE PARAMETERS IS A SCALAR, IT NEED NOT     00417700
C           BE DIMENSIONED IN THE USER-S CALLING PROGRAM.                00417800
C                                                                       00417900
C           IF NONE OF THE ABOVE CHOICES (WITH ITOL, RTOL, AND ATOL      00418000
C           FIXED THROUGHOUT THE PROBLEM) IS SUITABLE, MORE GENERAL      00418100
C           ERROR CONTROLS CAN BE OBTAINED BY SUBSTITUTING A             00418200
C           USER-SUPPLIED ROUTINE FOR THE SETTING OF EWT.                00418300
C           SEE PART IV BELOW.                                           00418400
C                                                                       00418500
C           IF GLOBAL ERRORS ARE TO BE ESTIMATED BY MAKING A REPEATED    00418600
C           RUN ON THE SAME PROBLEM WITH SMALLER TOLERANCES, THEN ALL    00418700
C           COMPONENTS OF RTOL AND ATOL (I.E. OF EWT) SHOULD BE SCALED   00418800
C           DOWN UNIFORMLY.                                              00418900
C                                                                       00419000
C ITASK   = AN INDEX SPECIFYING THE TASK TO BE PERFORMED.                00419100
C           INPUT ONLY.  ITASK HAS THE FOLLOWING VALUES AND MEANINGS.    00419200
C           1  MEANS NORMAL COMPUTATION OF OUTPUT VALUES OF Y(T) AT       00419300
C              T = TOUT (BY OVERSHOOTING AND INTERPOLATING).             00419400
C           2  MEANS TAKE ONE STEP ONLY AND RETURN.                      00419500
C           3  MEANS STOP AT THE FIRST INTERNAL MESH POINT AT OR         00419600
C              BEYOND T = TOUT AND RETURN.                               00419700
C           4  MEANS NORMAL COMPUTATION OF OUTPUT VALUES OF Y(T) AT      00419800
C              T = TOUT BUT WITHOUT OVERSHOOTING T = TCRIT.              00419900
C              TCRIT MUST BE INPUT AS RWORK(1).  TCRIT MAY BE EQUAL TO   00420000
C              OR BEYOND TOUT, BUT NOT BEHIND IT IN THE DIRECTION OF     00420100
C              INTEGRATION.  THIS OPTION IS USEFUL IF THE PROBLEM        00420200
C              HAS A SINGULARITY AT OR BEYOND T = TCRIT.                 00420300
C           5  MEANS TAKE ONE STEP, WITHOUT PASSING TCRIT, AND RETURN.   00420400
C              TCRIT MUST BE INPUT AS RWORK(1).                          00420500
C                                                                       00420600
C           NOTE..  IF ITASK = 4 OR 5 AND THE SOLVER REACHES TCRIT       00420700
C           (WITHIN ROUNDOFF), IT WILL RETURN T = TCRIT (EXACTLY) TO     00420800
C           INDICATE THIS (UNLESS ITASK = 4 AND TOUT COMES BEFORE TCRIT, 00420900
C           IN WHICH CASE ANSWERS AT T = TOUT ARE RETURNED FIRST).       00421000
C                                                                       00421100
C ISTATE  = AN INDEX USED FOR INPUT AND OUTPUT TO SPECIFY THE            00421200
C           THE STATE OF THE CALCULATION.                                00421300
C                                                                       00421400
C           ON INPUT, THE VALUES OF ISTATE ARE AS FOLLOWS.               00421500
C           1  MEANS THIS IS THE FIRST CALL FOR THE PROBLEM              00421600
C              (INITIALIZATIONS WILL BE DONE).  SEE NOTE BELOW.          00421700
C           2  MEANS THIS IS NOT THE FIRST CALL, AND THE CALCULATION     00421800
C              IS TO CONTINUE NORMALLY, WITH NO CHANGE IN ANY INPUT      00421900
C              PARAMETERS EXCEPT POSSIBLY TOUT AND ITASK.                00422000
C              (IF ITOL, RTOL, AND/OR ATOL ARE CHANGED BETWEEN CALLS     00422100
C              WITH ISTATE = 2, THE NEW VALUES WILL BE USED BUT NOT      00422200
C              TESTED FOR LEGALITY.)                                     00422300
C           3  MEANS THIS IS NOT THE FIRST CALL, AND THE                 00422400
C              CALCULATION IS TO CONTINUE NORMALLY, BUT WITH             00422500
C              A CHANGE IN INPUT PARAMETERS OTHER THAN                   00422600
C              TOUT AND ITASK.  CHANGES ARE ALLOWED IN                   00422700
C              NEQ, ITOL, RTOL, ATOL, IOPT, LRW, LIW, JT, ML, MU,        00422800
```

```
C             AND ANY OPTIONAL INPUTS EXCEPT HO, MXORDN, AND MXORDS.     00422900
C             (SEE IWORK DESCRIPTION FOR ML AND MU.)                     00423000
C             IN ADDITION, IMMEDIATELY FOLLOWING A RETURN WITH           00423100
C             ISTATE = 3 (ROOT FOUND), NG AND G MAY BE CHANGED.          00423200
C             (BUT CHANGING NG FROM O TO .GT. O IS NOT ALLOWED.)         00423300
C           NOTE..  A PRELIMINARY CALL WITH TOUT = T IS NOT COUNTED      00423400
C           AS A FIRST CALL HERE, AS NO INITIALIZATION OR CHECKING OF    00423500
C           INPUT IS DONE.  (SUCH A CALL IS SOMETIMES USEFUL FOR THE     00423600
C           PURPOSE OF OUTPUTTING THE INITIAL CONDITIONS.)               00423700
C           THUS THE FIRST CALL FOR WHICH TOUT .NE. T REQUIRES           00423800
C           ISTATE = 1 ON INPUT.                                         00423900
C                                                                        00424000
C           ON OUTPUT, ISTATE HAS THE FOLLOWING VALUES AND MEANINGS.     00424100
C            1  MEANS NOTHING WAS DONE, AS TOUT WAS EQUAL TO T WITH      00424200
C               ISTATE = 1 ON INPUT.  (HOWEVER, AN INTERNAL COUNTER WAS  00424300
C               SET TO DETECT AND PREVENT REPEATED CALLS OF THIS TYPE.)  00424400
C            2  MEANS THE INTEGRATION WAS PERFORMED SUCCESSFULLY, AND     00424500
C               NO ROOTS WERE FOUND.                                     00424600
C            3  MEANS THE INTEGRATION WAS SUCCESSFUL, AND ONE OR MORE     00424700
C               ROOTS WERE FOUND BEFORE SATISFYING THE STOP CONDITION    00424800
C               SPECIFIED BY ITASK.  SEE JROOT.                          00424900
C           -1  MEANS AN EXCESSIVE AMOUNT OF WORK (MORE THAN MXSTEP      00425000
C               STEPS) WAS DONE ON THIS CALL, BEFORE COMPLETING THE      00425100
C               REQUESTED TASK, BUT THE INTEGRATION WAS OTHERWISE        00425200
C               SUCCESSFUL AS FAR AS T.  (MXSTEP IS AN OPTIONAL INPUT    00425300
C               AND IS NORMALLY 500.)  TO CONTINUE, THE USER MAY         00425400
C               SIMPLY RESET ISTATE TO A VALUE .GT. 1 AND CALL AGAIN     00425500
C               (THE EXCESS WORK STEP COUNTER WILL BE RESET TO O).       00425600
C               IN ADDITION, THE USER MAY INCREASE MXSTEP TO AVOID       00425700
C               THIS ERROR RETURN (SEE BELOW ON OPTIONAL INPUTS).        00425800
C           -2  MEANS TOO MUCH ACCURACY WAS REQUESTED FOR THE PRECISION  00425900
C               OF THE MACHINE BEING USED.  THIS WAS DETECTED BEFORE     00426000
C               COMPLETING THE REQUESTED TASK, BUT THE INTEGRATION       00426100
C               WAS SUCCESSFUL AS FAR AS T.  TO CONTINUE, THE TOLERANCE  00426200
C               PARAMETERS MUST BE RESET, AND ISTATE MUST BE SET         00426300
C               TO 3.  THE OPTIONAL OUTPUT TOLSF MAY BE USED FOR THIS    00426400
C               PURPOSE.  (NOTE.. IF THIS CONDITION IS DETECTED BEFORE   00426500
C               TAKING ANY STEPS, THEN AN ILLEGAL INPUT RETURN           00426600
C               (ISTATE = -3) OCCURS INSTEAD.)                           00426700
C           -3  MEANS ILLEGAL INPUT WAS DETECTED, BEFORE TAKING ANY      00426800
C               INTEGRATION STEPS.  SEE WRITTEN MESSAGE FOR DETAILS.     00426900
C               NOTE..  IF THE SOLVER DETECTS AN INFINITE LOOP OF CALLS  00427000
C               TO THE SOLVER WITH ILLEGAL INPUT, IT WILL CAUSE          00427100
C               THE RUN TO STOP.                                         00427200
C           -4  MEANS THERE WERE REPEATED ERROR TEST FAILURES ON         00427300
C               ONE ATTEMPTED STEP, BEFORE COMPLETING THE REQUESTED      00427400
C               TASK, BUT THE INTEGRATION WAS SUCCESSFUL AS FAR AS T.    00427500
C               THE PROBLEM MAY HAVE A SINGULARITY, OR THE INPUT         00427600
C               MAY BE INAPPROPRIATE.                                    00427700
C           -5  MEANS THERE WERE REPEATED CONVERGENCE TEST FAILURES ON   00427800
C               ONE ATTEMPTED STEP, BEFORE COMPLETING THE REQUESTED      00427900
C               TASK, BUT THE INTEGRATION WAS SUCCESSFUL AS FAR AS T.    00428000
C               THIS MAY BE CAUSED BY AN INACCURATE JACOBIAN MATRIX,     00428100
C               IF ONE IS BEING USED.                                    00428200
C           -6  MEANS EWT(I) BECAME ZERO FOR SOME I DURING THE           00428300
C               INTEGRATION.  PURE RELATIVE ERROR CONTROL (ATOL(I)=O.O)  00428400
C               WAS REQUESTED ON A VARIABLE WHICH HAS NOW VANISHED.      00428500
C               THE INTEGRATION WAS SUCCESSFUL AS FAR AS T.              00428600
C           -7  MEANS THE LENGTH OF RWORK AND/OR IWORK WAS TOO SMALL TO  00428700
C               PROCEED, BUT THE INTEGRATION WAS SUCCESSFUL AS FAR AS T. 00428800
C               THIS HAPPENS WHEN LSODAR CHOOSES TO SWITCH METHODS       00428900
C               BUT LRW AND/OR LIW IS TOO SMALL FOR THE NEW METHOD.      00429000
C                                                                        00429100
C           NOTE..  SINCE THE NORMAL OUTPUT VALUE OF ISTATE IS 2,        00429200
C           IT DOES NOT NEED TO BE RESET FOR NORMAL CONTINUATION.        00429300
C           ALSO, SINCE A NEGATIVE INPUT VALUE OF ISTATE WILL BE         00429400
C           REGARDED AS ILLEGAL, A NEGATIVE OUTPUT VALUE REQUIRES THE    00429500
C           USER TO CHANGE IT, AND POSSIBLY OTHER INPUTS, BEFORE         00429600
C           CALLING THE SOLVER AGAIN.                                    00429700
C                                                                        00429800
C IOPT    = AN INTEGER FLAG TO SPECIFY WHETHER OR NOT ANY OPTIONAL       00429900
C           INPUTS ARE BEING USED ON THIS CALL.  INPUT ONLY.             00430000
```

```
C          THE OPTIONAL INPUTS ARE LISTED SEPARATELY BELOW.                    00430100
C          IOPT = 0 MEANS NO OPTIONAL INPUTS ARE BEING USED.                   00430200
C                   DEFAULT VALUES WILL BE USED IN ALL CASES.                  00430300
C          IOPT = 1 MEANS ONE OR MORE OPTIONAL INPUTS ARE BEING USED.          00430400
C                                                                              00430500
C RWORK  = A REAL ARRAY (DOUBLE PRECISION) FOR WORK SPACE, AND (IN THE         00430600
C          FIRST 20 WORDS) FOR CONDITIONAL AND OPTIONAL INPUTS AND             00430700
C          OPTIONAL OUTPUTS.                                                   00430800
C          AS LSODAR SWITCHES AUTOMATICALLY BETWEEN STIFF AND NONSTIFF         00430900
C          METHODS, THE REQUIRED LENGTH OF RWORK CAN CHANGE DURING THE         00431000
C          PROBLEM.  THUS THE RWORK ARRAY PASSED TO LSODAR CAN EITHER          00431100
C          HAVE A STATIC (FIXED) LENGTH LARGE ENOUGH FOR BOTH METHODS,         00431200
C          OR HAVE A DYNAMIC (CHANGING) LENGTH ALTERED BY THE CALLING          00431300
C          PROGRAM IN RESPONSE TO OUTPUT FROM LSODAR.                          00431400
C                                                                              00431500
C                      --- FIXED LENGTH CASE ---                               00431600
C          IF THE RWORK LENGTH IS TO BE FIXED, IT SHOULD BE AT LEAST           00431700
C               MAX (LRN, LRS),                                                00431800
C          WHERE LRN AND LRS ARE THE RWORK LENGTHS REQUIRED WHEN THE           00431900
C          CURRENT METHOD IS NONSTIFF OR STIFF, RESPECTIVELY.                  00432000
C                                                                              00432100
C          THE SEPARATE RWORK LENGTH REQUIREMENTS LRN AND LRS ARE              00432200
C          AS FOLLOWS..                                                        00432300
C          IF NEQ IS CONSTANT AND THE MAXIMUM METHOD ORDERS HAVE               00432400
C          THEIR DEFAULT VALUES, THEN                                          00432500
C             LRN = 20 + 16*NEQ + 3*NG,                                        00432600
C             LRS = 22 + 9*NEQ + NEQ**2 + 3*NG          (JT = 1 OR 2),         00432700
C             LRS = 22 + 10*NEQ + (2*ML+MU)*NEQ + 3*NG  (JT = 4 OR 5).         00432800
C          UNDER ANY OTHER CONDITIONS, LRN AND LRS ARE GIVEN BY..              00432900
C             LRN = 20 + NYH*(MXORDN+1) + 3*NEQ + 3*NG,                        00433000
C             LRS = 20 + NYH*(MXORDS+1) + 3*NEQ + LMAT + 3*NG,                 00433100
C          WHERE                                                               00433200
C             NYH    = THE INITIAL VALUE OF NEQ,                               00433300
C             MXORDN = 12, UNLESS A SMALLER VALUE IS GIVEN AS AN               00433400
C                      OPTIONAL INPUT,                                         00433500
C             MXORDS = 5, UNLESS A SMALLER VALUE IS GIVEN AS AN                00433600
C                      OPTIONAL INPUT,                                         00433700
C             LMAT   = LENGTH OF MATRIX WORK SPACE..                           00433800
C             LMAT   = NEQ**2 + 2              IF JT = 1 OR 2,                 00433900
C             LMAT   = (2*ML + MU + 1)*NEQ + 2 IF JT = 4 OR 5.                 00434000
C                                                                              00434100
C                      --- DYNAMIC LENGTH CASE ---                             00434200
C          IF THE LENGTH OF RWORK IS TO BE DYNAMIC, THEN IT SHOULD             00434300
C          BE AT LEAST LRN OR LRS, AS DEFINED ABOVE, DEPENDING ON THE          00434400
C          CURRENT METHOD.  INITIALLY, IT MUST BE AT LEAST LRN (SINCE          00434500
C          LSODAR STARTS WITH THE NONSTIFF METHOD).  ON ANY RETURN             00434600
C          FROM LSODAR, THE OPTIONAL OUTPUT MCUR INDICATES THE CURRENT         00434700
C          METHOD.  IF MCUR DIFFERS FROM THE VALUE IT HAD ON THE               00434800
C          PREVIOUS RETURN, OR IF THERE HAS ONLY BEEN ONE CALL TO              00434900
C          LSODAR AND MCUR IS NOW 2, THEN LSODAR HAS SWITCHED                  00435000
C          METHODS DURING THE LAST CALL, AND THE LENGTH OF RWORK               00435100
C          SHOULD BE RESET (TO LRN IF MCUR = 1, OR TO LRS IF                   00435200
C          MCUR = 2).  (AN INCREASE IN THE RWORK LENGTH IS REQUIRED            00435300
C          IF LSODAR RETURNED ISTATE = -7, BUT NOT OTHERWISE.)                 00435400
C          AFTER RESETTING THE LENGTH, CALL LSODAR WITH ISTATE = 3             00435500
C          TO SIGNAL THAT CHANGE.                                              00435600
C                                                                              00435700
C LRW    = THE LENGTH OF THE ARRAY RWORK, AS DECLARED BY THE USER.             00435800
C          (THIS WILL BE CHECKED BY THE SOLVER.)                               00435900
C                                                                              00436000
C IWORK  = AN INTEGER ARRAY FOR WORK SPACE.                                    00436100
C          AS LSODAR SWITCHES AUTOMATICALLY BETWEEN STIFF AND NONSTIFF         00436200
C          METHODS, THE REQUIRED LENGTH OF IWORK CAN CHANGE DURING             00436300
C          PROBLEM, BETWEEN                                                    00436400
C             LIS = 20 + NEQ   AND   LIN = 20,                                 00436500
C          RESPECTIVELY.  THUS THE IWORK ARRAY PASSED TO LSODAR CAN            00436600
C          EITHER HAVE A FIXED LENGTH OF AT LEAST 20 + NEQ, OR HAVE A          00436700
C          DYNAMIC LENGTH OF AT LEAST LIN OR LIS, DEPENDING ON THE             00436800
C          CURRENT METHOD.  THE COMMENTS ON DYNAMIC LENGTH UNDER               00436900
C          RWORK ABOVE APPLY HERE.  INITIALLY, THIS LENGTH NEED                00437000
C          ONLY BE AT LEAST LIN = 20.                                          00437100
C                                                                              00437200
```

```
C          THE FIRST FEW WORDS OF IWORK ARE USED FOR CONDITIONAL AND    00437300
C          OPTIONAL INPUTS AND OPTIONAL OUTPUTS.                        00437400
C                                                                       00437500
C          THE FOLLOWING 2 WORDS IN IWORK ARE CONDITIONAL INPUTS..      00437600
C            IWORK(1) = ML     THESE ARE THE LOWER AND UPPER            00437700
C            IWORK(2) = MU     HALF-BANDWIDTHS, RESPECTIVELY, OF THE    00437800
C                       BANDED JACOBIAN, EXCLUDING THE MAIN DIAGONAL.   00437900
C                       THE BAND IS DEFINED BY THE MATRIX LOCATIONS     00438000
C                       (I,J) WITH I-ML .LE. J .LE. I+MU.  ML AND MU    00438100
C                       MUST SATISFY  0 .LE.  ML,MU  .LE. NEQ-1.        00438200
C                       THESE ARE REQUIRED IF JT IS 4 OR 5, AND         00438300
C                       IGNORED OTHERWISE.  ML AND MU MAY IN FACT BE    00438400
C                       THE BAND PARAMETERS FOR A MATRIX TO WHICH       00438500
C                       DF/DY IS ONLY APPROXIMATELY EQUAL.              00438600
C                                                                       00438700
C LIW    = THE LENGTH OF THE ARRAY IWORK, AS DECLARED BY THE USER.      00438800
C          (THIS WILL BE CHECKED BY THE SOLVER.)                        00438900
C                                                                       00439000
C NOTE.. THE BASE ADDRESSES OF THE WORK ARRAYS MUST NOT BE              00439100
C ALTERED BETWEEN CALLS TO LSODAR FOR THE SAME PROBLEM.                 00439200
C THE CONTENTS OF THE WORK ARRAYS MUST NOT BE ALTERED                   00439300
C BETWEEN CALLS, EXCEPT POSSIBLY FOR THE CONDITIONAL AND                00439400
C OPTIONAL INPUTS, AND EXCEPT FOR THE LAST 3*NEQ WORDS OF RWORK.        00439500
C THE LATTER SPACE IS USED FOR INTERNAL SCRATCH SPACE, AND SO IS        00439600
C AVAILABLE FOR USE BY THE USER OUTSIDE LSODAR BETWEEN CALLS, IF        00439700
C DESIRED (BUT NOT FOR USE BY F, JAC, OR G).                            00439800
C                                                                       00439900
C JAC    = THE NAME OF THE USER-SUPPLIED ROUTINE TO COMPUTE THE         00440000
C          JACOBIAN MATRIX, DF/DY, IF JT = 1 OR 4.  THE JAC ROUTINE     00440100
C          IS OPTIONAL, BUT IF THE PROBLEM IS EXPECTED TO BE STIFF MUCH 00440200
C          OF THE TIME, YOU ARE ENCOURAGED TO SUPPLY JAC, FOR THE SAKE  00440300
C          OF EFFICIENCY.  (ALTERNATIVELY, SET JT = 2 OR 5 TO HAVE      00440400
C          LSODAR COMPUTE DF/DY INTERNALLY BY DIFFERENCE QUOTIENTS.)    00440500
C          IF AND WHEN LSODAR USES DF/DY, IF TREATS THIS NEQ BY NEQ     00440600
C          MATRIX EITHER AS FULL (JT = 1 OR 2), OR AS BANDED (JT =      00440700
C          4 OR 5) WITH HALF-BANDWIDTHS ML AND MU (DISCUSSED UNDER      00440800
C          IWORK ABOVE).  IN EITHER CASE, IF JT = 1 OR 4, THE JAC       00440900
C          ROUTINE MUST COMPUTE DF/DY AS A FUNCTION OF THE SCALAR T     00441000
C          AND THE VECTOR Y.  IT IS TO HAVE THE FORM                    00441100
C               SUBROUTINE JAC (NEQ, T, Y, ML, MU, PD, NROWPD)          00441200
C               DIMENSION Y(1), PD(NROWPD,1)                            00441300
C          WHERE NEQ, T, Y, ML, MU, AND NROWPD ARE INPUT AND THE ARRAY  00441400
C          PD IS TO BE LOADED WITH PARTIAL DERIVATIVES (ELEMENTS OF     00441500
C          THE JACOBIAN MATRIX) ON OUTPUT.  PD MUST BE GIVEN A FIRST    00441600
C          DIMENSION OF NROWPD.  T AND Y HAVE THE SAME MEANING AS IN    00441700
C          SUBROUTINE F.  (IN THE DIMENSION STATEMENT ABOVE, 1 IS A     00441800
C          DUMMY DIMENSION.. IT CAN BE REPLACED BY ANY VALUE.)          00441900
C               IN THE FULL MATRIX CASE (JT = 1), ML AND MU ARE         00442000
C          IGNORED, AND THE JACOBIAN IS TO BE LOADED INTO PD IN         00442100
C          COLUMNWISE MANNER, WITH DF(I)/DY(J) LOADED INTO PD(I,J).     00442200
C               IN THE BAND MATRIX CASE (JT = 4), THE ELEMENTS          00442300
C          WITHIN THE BAND ARE TO BE LOADED INTO PD IN COLUMNWISE       00442400
C          MANNER, WITH DIAGONAL LINES OF DF/DY LOADED INTO THE ROWS    00442500
C          OF PD.  THUS DF(I)/DY(J) IS TO BE LOADED INTO PD(I-J+MU+1,J).00442600
C          ML AND MU ARE THE HALF-BANDWIDTH PARAMETERS (SEE IWORK).     00442700
C          THE LOCATIONS IN PD IN THE TWO TRIANGULAR AREAS WHICH        00442800
C          CORRESPOND TO NONEXISTENT MATRIX ELEMENTS CAN BE IGNORED     00442900
C          OR LOADED ARBITRARILY, AS THEY ARE OVERWRITTEN BY LSODAR.    00443000
C               JAC NEED NOT PROVIDE DF/DY EXACTLY.  A CRUDE            00443100
C          APPROXIMATION (POSSIBLY WITH A SMALLER BANDWIDTH) WILL DO.   00443200
C               IN EITHER CASE, PD IS PRESET TO ZERO BY THE SOLVER,     00443300
C          SO THAT ONLY THE NONZERO ELEMENTS NEED BE LOADED BY JAC.     00443400
C          EACH CALL TO JAC IS PRECEDED BY A CALL TO F WITH THE SAME    00443500
C          ARGUMENTS NEQ, T, AND Y.  THUS TO GAIN SOME EFFICIENCY,      00443600
C          INTERMEDIATE QUANTITIES SHARED BY BOTH CALCULATIONS MAY BE   00443700
C          SAVED IN A USER COMMON BLOCK BY F AND NOT RECOMPUTED BY JAC, 00443800
C          IF DESIRED.  ALSO, JAC MAY ALTER THE Y ARRAY, IF DESIRED.    00443900
C          JAC MUST BE DECLARED EXTERNAL IN THE CALLING PROGRAM.        00444000
C               SUBROUTINE JAC MAY ACCESS USER-DEFINED QUANTITIES IN    00444100
C          NEQ(2),... AND Y(NEQ(1)+1),... IF NEQ IS AN ARRAY            00444200
C          (DIMENSIONED IN JAC) AND Y HAS LENGTH EXCEEDING NEQ(1).      00444300
C          SEE THE DESCRIPTIONS OF NEQ AND Y ABOVE.                     00444400
```

```
C                                                                       00444500
C JT      = JACOBIAN TYPE INDICATOR.  USED ONLY FOR INPUT.              00444600
C           JT SPECIFIES HOW THE JACOBIAN MATRIX DF/DY WILL BE          00444700
C           TREATED, IF AND WHEN LSODAR REQUIRES THIS MATRIX.           00444800
C           JT HAS THE FOLLOWING VALUES AND MEANINGS..                  00444900
C            1 MEANS A USER-SUPPLIED FULL (NEQ BY NEQ) JACOBIAN.        00445000
C            2 MEANS AN INTERNALLY GENERATED (DIFFERENCE QUOTIENT) FULL  00445100
C              JACOBIAN (USING NEQ EXTRA CALLS TO F PER DF/DY VALUE).   00445200
C            4 MEANS A USER-SUPPLIED BANDED JACOBIAN.                   00445300
C            5 MEANS AN INTERNALLY GENERATED BANDED JACOBIAN (USING     00445400
C              ML+MU+1 EXTRA CALLS TO F PER DF/DY EVALUATION).          00445500
C           IF JT = 1 OR 4, THE USER MUST SUPPLY A SUBROUTINE JAC       00445600
C           (THE NAME IS ARBITRARY) AS DESCRIBED ABOVE UNDER JAC.       00445700
C           IF JT = 2 OR 5, A DUMMY ARGUMENT CAN BE USED.               00445800
C                                                                       00445900
C G       = THE NAME OF SUBROUTINE FOR CONSTRAINT FUNCTIONS, WHOSE      00446000
C           ROOTS ARE DESIRED DURING THE INTEGRATION.  IT IS TO HAVE    00446100
C           THE FORM                                                    00446200
C                SUBROUTINE G (NEQ, T, Y, NG, GOUT)                     00446300
C                DIMENSION Y(NEQ), GOUT(NG)                             00446400
C           WHERE NEQ, T, Y, AND NG ARE INPUT, AND THE ARRAY GOUT       00446500
C           IS OUTPUT.  NEQ, T, AND Y HAVE THE SAME MEANING AS IN       00446600
C           THE F ROUTINE, AND GOUT IS AN ARRAY OF LENGTH NG.           00446700
C           FOR I = 1,...,NG, THIS ROUTINE IS TO LOAD INTO GOUT(I)      00446800
C           THE VALUE AT (T,Y) OF THE I-TH CONSTRAINT FUNCTION G(I).    00446900
C           LSODAR WILL FIND ROOTS OF THE G(I) OF ODD MULTIPLICITY      00447000
C           (I.E. SIGN CHANGES) AS THEY OCCUR DURING THE INTEGRATION.   00447100
C           G MUST BE DECLARED EXTERNAL IN THE CALLING PROGRAM.         00447200
C                                                                       00447300
C           CAUTION.. BECAUSE OF NUMERICAL ERRORS IN THE FUNCTIONS      00447400
C           G(I) DUE TO ROUNDOFF AND INTEGRATION ERROR, LSODAR MAY      00447500
C           RETURN FALSE ROOTS, OR RETURN THE SAME ROOT AT TWO OR MORE  00447600
C           NEARLY EQUAL VALUES OF T.  IF SUCH FALSE ROOTS ARE          00447700
C           SUSPECTED, THE USER SHOULD CONSIDER SMALLER ERROR TOLERANCES 00447800
C           AND/OR HIGHER PRECISION IN THE EVALUATION OF THE G(I).      00447900
C                                                                       00448000
C           IF A ROOT OF SOME G(I) DEFINES THE END OF THE PROBLEM,      00448100
C           THE INPUT TO LSODAR SHOULD NEVERTHELESS ALLOW INTEGRATION   00448200
C           TO A POINT SLIGHTLY PAST THAT ROOT, SO THAT LSODAR CAN      00448300
C           LOCATE THE ROOT BY INTERPOLATION.                           00448400
C                                                                       00448500
C           SUBROUTINE G MAY ACCESS USER-DEFINED QUANTITIES IN          00448600
C           NEQ(2),... AND Y(NEQ(1)+1),... IF NEQ IS AN ARRAY           00448700
C           (DIMENSIONED IN G) AND Y HAS LENGTH EXCEEDING NEQ(1).       00448800
C           SEE THE DESCRIPTIONS OF NEQ AND Y ABOVE.                    00448900
C                                                                       00449000
C NG      = NUMBER OF CONSTRAINT FUNCTIONS G(I).  IF THERE ARE NONE,    00449100
C           SET NG = 0, AND PASS A DUMMY NAME FOR G.                    00449200
C                                                                       00449300
C JROOT   = INTEGER ARRAY OF LENGTH NG.  USED ONLY FOR OUTPUT.          00449400
C           ON A RETURN WITH ISTATE = 3 (ONE OR MORE ROOTS FOUND),      00449500
C           JROOT(I) = 1 IF G(I) HAS A ROOT AT T, OR JROOT(I) = 0 IF NOT.00449600
C-----------------------------------------------------------------------00449700
C OPTIONAL INPUTS.                                                      00449800
C                                                                       00449900
C THE FOLLOWING IS A LIST OF THE OPTIONAL INPUTS PROVIDED FOR IN THE    00450000
C CALL SEQUENCE.  (SEE ALSO PART II.)  FOR EACH SUCH INPUT VARIABLE,    00450100
C THIS TABLE LISTS ITS NAME AS USED IN THIS DOCUMENTATION, ITS          00450200
C LOCATION IN THE CALL SEQUENCE, ITS MEANING, AND THE DEFAULT VALUE.    00450300
C THE USE OF ANY OF THESE INPUTS REQUIRES IOPT = 1, AND IN THAT         00450400
C CASE ALL OF THESE INPUTS ARE EXAMINED.  A VALUE OF ZERO FOR ANY       00450500
C OF THESE OPTIONAL INPUTS WILL CAUSE THE DEFAULT VALUE TO BE USED.     00450600
C THUS TO USE A SUBSET OF THE OPTIONAL INPUTS, SIMPLY PRELOAD           00450700
C LOCATIONS 5 TO 10 IN RWORK AND IWORK TO 0.0 AND 0 RESPECTIVELY, AND   00450800
C THEN SET THOSE OF INTEREST TO NONZERO VALUES.                         00450900
C                                                                       00451000
C NAME    LOCATION      MEANING AND DEFAULT VALUE                       00451100
C                                                                       00451200
C HO      RWORK(5)  THE STEP SIZE TO BE ATTEMPTED ON THE FIRST STEP.    00451300
C                   THE DEFAULT VALUE IS DETERMINED BY THE SOLVER.      00451400
C                                                                       00451500
C HMAX    RWORK(6)  THE MAXIMUM ABSOLUTE STEP SIZE ALLOWED.             00451600
```

```
C                     THE DEFAULT VALUE IS INFINITE.                        00451700
C                                                                           00451800
C HMIN      RWORK(7)  THE MINIMUM ABSOLUTE STEP SIZE ALLOWED.               00451900
C                     THE DEFAULT VALUE IS 0.  (THIS LOWER BOUND IS NOT     00452000
C                     ENFORCED ON THE FINAL STEP BEFORE REACHING TCRIT     00452100
C                     WHEN ITASK = 4 OR 5.)                                 00452200
C                                                                           00452300
C IXPR      IWORK(5)  FLAG TO GENERATE EXTRA PRINTING AT METHOD SWITCHES.   00452400
C                     IXPR = 0 MEANS NO EXTRA PRINTING (THE DEFAULT).       00452500
C                     IXPR = 1 MEANS PRINT DATA ON EACH SWITCH.             00452600
C                     T, H, AND NST WILL BE PRINTED ON THE SAME LOGICAL     00452700
C                     UNIT AS USED FOR ERROR MESSAGES.                      00452800
C                                                                           00452900
C MXSTEP    IWORK(6)  MAXIMUM NUMBER OF (INTERNALLY DEFINED) STEPS          00453000
C                     ALLOWED DURING ONE CALL TO THE SOLVER.                00453100
C                     THE DEFAULT VALUE IS 500.                             00453200
C                                                                           00453300
C MXHNIL    IWORK(7)  MAXIMUM NUMBER OF MESSAGES PRINTED (PER PROBLEM)      00453400
C                     WARNING THAT T + H = T ON A STEP (H = STEP SIZE).     00453500
C                     THIS MUST BE POSITIVE TO RESULT IN A NON-DEFAULT      00453600
C                     VALUE.  THE DEFAULT VALUE IS 10.                      00453700
C                                                                           00453800
C MXORDN    IWORK(8)  THE MAXIMUM ORDER TO BE ALLOWED FOR THE NONSTIFF      00453900
C                     (ADAMS) METHOD.  THE DEFAULT VALUE IS 12.             00454000
C                     IF MXORDN EXCEEDS THE DEFAULT VALUE, IT WILL          00454100
C                     BE REDUCED TO THE DEFAULT VALUE.                      00454200
C                     MXORDN IS HELD CONSTANT DURING THE PROBLEM.           00454300
C                                                                           00454400
C MXORDS    IWORK(9)  THE MAXIMUM ORDER TO BE ALLOWED FOR THE STIFF         00454500
C                     (BDF) METHOD.  THE DEFAULT VALUE IS 5.                00454600
C                     IF MXORDS EXCEEDS THE DEFAULT VALUE, IT WILL          00454700
C                     BE REDUCED TO THE DEFAULT VALUE.                      00454800
C                     MXORDS IS HELD CONSTANT DURING THE PROBLEM.           00454900
C-----------------------------------------------------------------------    00455000
C OPTIONAL OUTPUTS.                                                         00455100
C                                                                           00455200
C AS OPTIONAL ADDITIONAL OUTPUT FROM LSODAR, THE VARIABLES LISTED           00455300
C BELOW ARE QUANTITIES RELATED TO THE PERFORMANCE OF LSODAR                 00455400
C WHICH ARE AVAILABLE TO THE USER.  THESE ARE COMMUNICATED BY WAY OF        00455500
C THE WORK ARRAYS, BUT ALSO HAVE INTERNAL MNEMONIC NAMES AS SHOWN.          00455600
C EXCEPT WHERE STATED OTHERWISE, ALL OF THESE OUTPUTS ARE DEFINED           00455700
C ON ANY SUCCESSFUL RETURN FROM LSODAR, AND ON ANY RETURN WITH              00455800
C ISTATE = -1, -2, -4, -5, OR -6.  ON AN ILLEGAL INPUT RETURN               00455900
C (ISTATE = -3), THEY WILL BE UNCHANGED FROM THEIR EXISTING VALUES          00456000
C (IF ANY), EXCEPT POSSIBLY FOR TOLSF, LENRW, AND LENIW.                    00456100
C ON ANY ERROR RETURN, OUTPUTS RELEVANT TO THE ERROR WILL BE DEFINED,       00456200
C AS NOTED BELOW.                                                           00456300
C                                                                           00456400
C NAME      LOCATION      MEANING                                           00456500
C                                                                           00456600
C HU        RWORK(11) THE STEP SIZE IN T LAST USED (SUCCESSFULLY).          00456700
C                                                                           00456800
C HCUR      RWORK(12) THE STEP SIZE TO BE ATTEMPTED ON THE NEXT STEP.       00456900
C                                                                           00457000
C TCUR      RWORK(13) THE CURRENT VALUE OF THE INDEPENDENT VARIABLE         00457100
C                     WHICH THE SOLVER HAS ACTUALLY REACHED, I.E. THE       00457200
C                     CURRENT INTERNAL MESH POINT IN T.  ON OUTPUT, TCUR    00457300
C                     WILL ALWAYS BE AT LEAST AS FAR AS THE ARGUMENT        00457400
C                     T, BUT MAY BE FARTHER (IF INTERPOLATION WAS DONE).    00457500
C                                                                           00457600
C TOLSF     RWORK(14) A TOLERANCE SCALE FACTOR, GREATER THAN 1.0,           00457700
C                     COMPUTED WHEN A REQUEST FOR TOO MUCH ACCURACY WAS     00457800
C                     DETECTED (ISTATE = -3 IF DETECTED AT THE START OF     00457900
C                     THE PROBLEM, ISTATE = -2 OTHERWISE).  IF ITOL IS      00458000
C                     LEFT UNALTERED BUT RTOL AND ATOL ARE UNIFORMLY        00458100
C                     SCALED UP BY A FACTOR OF TOLSF FOR THE NEXT CALL,     00458200
C                     THEN THE SOLVER IS DEEMED LIKELY TO SUCCEED.          00458300
C                     (THE USER MAY ALSO IGNORE TOLSF AND ALTER THE         00458400
C                     TOLERANCE PARAMETERS IN ANY OTHER WAY APPROPRIATE.)   00458500
C                                                                           00458600
C TSW       RWORK(15) THE VALUE OF T AT THE TIME OF THE LAST METHOD         00458700
C                     SWITCH, IF ANY.                                       00458800
```

```
C                                                                       00458900
C NGE      IWORK(10) THE NUMBER OF G EVALUATIONS FOR THE PROBLEM SO FAR. 00459000
C                                                                       00459100
C NST      IWORK(11) THE NUMBER OF STEPS TAKEN FOR THE PROBLEM SO FAR.  00459200
C                                                                       00459300
C NFE      IWORK(12) THE NUMBER OF F EVALUATIONS FOR THE PROBLEM SO FAR. 00459400
C                                                                       00459500
C NJE      IWORK(13) THE NUMBER OF JACOBIAN EVALUATIONS (AND OF MATRIX  00459600
C                    LU DECOMPOSITIONS) FOR THE PROBLEM SO FAR.         00459700
C                                                                       00459800
C NQU      IWORK(14) THE METHOD ORDER LAST USED (SUCCESSFULLY).         00459900
C                                                                       00460000
C NQCUR    IWORK(15) THE ORDER TO BE ATTEMPTED ON THE NEXT STEP.        00460100
C                                                                       00460200
C IMXER    IWORK(16) THE INDEX OF THE COMPONENT OF LARGEST MAGNITUDE IN  00460300
C                    THE WEIGHTED LOCAL ERROR VECTOR ( E(I)/EWT(I) ),   00460400
C                    ON AN ERROR RETURN WITH ISTATE = -4 OR -5.         00460500
C                                                                       00460600
C LENRW    IWORK(17) THE LENGTH OF RWORK ACTUALLY REQUIRED, ASSUMING    00460700
C                    THAT THE LENGTH OF RWORK IS TO BE FIXED FOR THE    00460800
C                    REST OF THE PROBLEM, AND THAT SWITCHING MAY OCCUR. 00460900
C                    THIS IS DEFINED ON NORMAL RETURNS AND ON AN ILLEGAL 00461000
C                    INPUT RETURN FOR INSUFFICIENT STORAGE.             00461100
C                                                                       00461200
C LENIW    IWORK(18) THE LENGTH OF IWORK ACTUALLY REQUIRED, ASSUMING    00461300
C                    THAT THE LENGTH OF IWORK IS TO BE FIXED FOR THE    00461400
C                    REST OF THE PROBLEM, AND THAT SWITCHING MAY OCCUR. 00461500
C                    THIS IS DEFINED ON NORMAL RETURNS AND ON AN ILLEGAL 00461600
C                    INPUT RETURN FOR INSUFFICIENT STORAGE.             00461700
C                                                                       00461800
C MUSED    IWORK(19) THE METHOD INDICATOR FOR THE LAST SUCCESSFUL STEP.. 00461900
C                    1 MEANS ADAMS (NONSTIFF), 2 MEANS BDF (STIFF).     00462000
C                                                                       00462100
C MCUR     IWORK(20) THE CURRENT METHOD INDICATOR..                     00462200
C                    1 MEANS ADAMS (NONSTIFF), 2 MEANS BDF (STIFF).     00462300
C                    THIS IS THE METHOD TO BE ATTEMPTED                 00462400
C                    ON THE NEXT STEP.  THUS IT DIFFERS FROM MUSED      00462500
C                    ONLY IF A METHOD SWITCH HAS JUST BEEN MADE.        00462600
C                                                                       00462700
C THE FOLLOWING TWO ARRAYS ARE SEGMENTS OF THE RWORK ARRAY WHICH        00462800
C MAY ALSO BE OF INTEREST TO THE USER AS OPTIONAL OUTPUTS.              00462900
C FOR EACH ARRAY,. THE TABLE BELOW GIVES ITS INTERNAL NAME,             00463000
C ITS BASE ADDRESS IN RWORK, AND ITS DESCRIPTION.                       00463100
C                                                                       00463200
C NAME     BASE ADDRESS      DESCRIPTION                                00463300
C                                                                       00463400
C YH       21 + 3*NG       THE NORDSIECK HISTORY ARRAY, OF SIZE NYH BY  00463500
C                          (NQCUR + 1), WHERE NYH IS THE INITIAL VALUE  00463600
C                          OF NEQ.  FOR J = 0,1,...,NQCUR, COLUMN J+1   00463700
C                          OF YH CONTAINS HCUR**J/FACTORIAL(J) TIMES    00463800
C                          THE J-TH DERIVATIVE OF THE INTERPOLATING     00463900
C                          POLYNOMIAL CURRENTLY REPRESENTING THE SOLUTION,00464000
C                          EVALUATED AT T = TCUR.                       00464100
C                                                                       00464200
C ACOR      LACOR          ARRAY OF SIZE NEQ USED FOR THE ACCUMULATED   00464300
C          (FROM COMMON    CORRECTIONS ON EACH STEP, SCALED ON OUTPUT   00464400
C            AS NOTED)     TO REPRESENT THE ESTIMATED LOCAL ERROR IN Y  00464500
C                          ON THE LAST STEP.  THIS IS THE VECTOR E IN   00464600
C                          THE DESCRIPTION OF THE ERROR CONTROL.  IT IS 00464700
C                          DEFINED ONLY ON A SUCCESSFUL RETURN FROM     00464800
C                          LSODAR.  THE BASE ADDRESS LACOR IS OBTAINED BY 00464900
C                          INCLUDING IN THE USER-S PROGRAM THE          00465000
C                          FOLLOWING 3 LINES..                          00465100
C                             DOUBLE PRECISION RLS                      00465200
C                             COMMON /LS0001/ RLS(219), ILS(39)         00465300
C                             LACOR = ILS(5)                            00465400
C                                                                       00465500
C-----------------------------------------------------------------------00465600
C PART II.  OTHER ROUTINES CALLABLE.                                    00465700
C                                                                       00465800
C THE FOLLOWING ARE OPTIONAL CALLS WHICH THE USER MAY MAKE TO           00465900
C GAIN ADDITIONAL CAPABILITIES IN CONJUNCTION WITH LSODAR.              00466000
```

```
C (THE ROUTINES XSETUN AND XSETF ARE DESIGNED TO CONFORM TO THE          00466100
C SLATEC ERROR HANDLING PACKAGE.)                                        00466200
C                                                                        00466300
C     FORM OF CALL                  FUNCTION                             00466400
C   CALL XSETUN(LUN)           SET THE LOGICAL UNIT NUMBER, LUN, FOR     00466500
C                              OUTPUT OF MESSAGES FROM LSODAR, IF        00466600
C                              THE DEFAULT IS NOT DESIRED.               00466700
C                              THE DEFAULT VALUE OF LUN IS 6.            00466800
C                                                                        00466900
C   CALL XSETF(MFLAG)          SET A FLAG TO CONTROL THE PRINTING OF     00467000
C                              MESSAGES BY LSODAR.                       00467100
C                              MFLAG = 0 MEANS DO NOT PRINT. (DANGER..   00467200
C                              THIS RISKS LOSING VALUABLE INFORMATION.)  00467300
C                              MFLAG = 1 MEANS PRINT (THE DEFAULT).      00467400
C                                                                        00467500
C                              EITHER OF THE ABOVE CALLS MAY BE MADE AT  00467600
C                              ANY TIME AND WILL TAKE EFFECT IMMEDIATELY.00467700
C                                                                        00467800
C   CALL SVCAR (RSAV, ISAV)    STORE IN RSAV AND ISAV THE CONTENTS       00467900
C                              OF THE INTERNAL COMMON BLOCKS USED BY     00468000
C                              LSODAR (SEE PART III BELOW).              00468100
C                              RSAV MUST BE A REAL ARRAY OF LENGTH 246   00468200
C                              OR MORE, AND ISAV MUST BE AN INTEGER      00468300
C                              ARRAY OF LENGTH 59 OR MORE.               00468400
C                                                                        00468500
C   CALL RSCAR (RSAV, ISAV)    RESTORE, FROM RSAV AND ISAV, THE CONTENTS 00468600
C                              OF THE INTERNAL COMMON BLOCKS USED BY     00468700
C                              LSODAR.  PRESUMES A PRIOR CALL TO SVCAR   00468800
C                              WITH THE SAME ARGUMENTS.                  00468900
C                                                                        00469000
C                              SVCAR AND RSCAR ARE USEFUL IF             00469100
C                              INTERRUPTING A RUN AND RESTARTING         00469200
C                              LATER, OR ALTERNATING BETWEEN TWO OR      00469300
C                              MORE PROBLEMS SOLVED WITH LSODAR.         00469400
C                                                                        00469500
C   CALL INTDY(,,,,,)          PROVIDE DERIVATIVES OF Y, OF VARIOUS      00469600
C        (SEE BELOW)           ORDERS, AT A SPECIFIED,POINT T, IF        00469700
C                              DESIRED.  IT MAY BE CALLED ONLY AFTER     00469800
C                              A SUCCESSFUL RETURN FROM LSODAR.          00469900
C                                                                        00470000
C THE DETAILED INSTRUCTIONS FOR USING INTDY ARE AS FOLLOWS.              00470100
C THE FORM OF THE CALL IS..                                              00470200
C                                                                        00470300
C   CALL INTDY (T, K, RWORK(LYH), NYH, DKY, IFLAG)                       00470400
C                                                                        00470500
C THE INPUT PARAMETERS ARE..                                             00470600
C                                                                        00470700
C T         = VALUE OF INDEPENDENT VARIABLE WHERE ANSWERS ARE DESIRED    00470800
C             (NORMALLY THE SAME AS THE T LAST RETURNED BY LSODAR).      00470900
C             FOR VALID RESULTS, T MUST LIE BETWEEN TCUR - HU AND TCUR.  00471000
C             (SEE OPTIONAL OUTPUTS FOR TCUR AND HU.)                    00471100
C K         = INTEGER ORDER OF THE DERIVATIVE DESIRED.  K MUST SATISFY   00471200
C             0 .LE. K .LE. NQCUR, WHERE NQCUR IS THE CURRENT ORDER      00471300
C             (SEE OPTIONAL OUTPUTS).  THE CAPABILITY CORRESPONDING      00471400
C             TO K = 0, I.E. COMPUTING Y(T), IS ALREADY PROVIDED         00471500
C             BY LSODAR DIRECTLY.  SINCE NQCUR .GE. 1, THE FIRST         00471600
C             DERIVATIVE DY/DT IS ALWAYS AVAILABLE WITH INTDY.           00471700
C LYH       = 21 + 3*NG = BASE ADDRESS IN RWORK OF THE HISTORY ARRAY YH. 00471800
C NYH       = COLUMN LENGTH OF YH, EQUAL TO THE INITIAL VALUE OF NEQ.    00471900
C                                                                        00472000
C THE OUTPUT PARAMETERS ARE..                                            00472100
C                                                                        00472200
C DKY       = A REAL ARRAY OF LENGTH NEQ CONTAINING THE COMPUTED VALUE   00472300
C             OF THE K-TH DERIVATIVE OF Y(T).                            00472400
C IFLAG     = INTEGER FLAG, RETURNED AS 0 IF K AND T WERE LEGAL,         00472500
C             -1 IF K WAS ILLEGAL, AND -2 IF T WAS ILLEGAL.              00472600
C             ON AN ERROR RETURN, A MESSAGE IS ALSO WRITTEN.             00472700
C-----------------------------------------------------------------------00472800
C PART III.  COMMON BLOCKS.                                              00472900
C                                                                        00473000
C IF LSODAR IS TO BE USED IN AN OVERLAY SITUATION, THE USER              00473100
C MUST DECLARE, IN THE PRIMARY OVERLAY, THE VARIABLES IN..               00473200
```

```
C   (1) THE CALL SEQUENCE TO LSODAR,                                    00473300
C   (2) THE FOUR INTERNAL COMMON BLOCKS                                 00473400
C         /LS0001/  OF LENGTH  258  (219 DOUBLE PRECISION WORDS         00473500
C                        FOLLOWED BY 39 INTEGER WORDS),                 00473600
C         /LSA001/  OF LENGTH  31    (22 DOUBLE PRECISION WORDS         00473700
C                        FOLLOWED BY  9 INTEGER WORDS),                 00473800
C         /LSR001/  OF LENGTH  14     (5 DOUBLE PRECISION WORDS         00473900
C                        FOLLOWED BY  9 INTEGER WORDS),                 00474000
C         /EH0001/  OF LENGTH  2 (INTEGER WORDS).                       00474100
C                                                                       00474200
C IF LSODAR IS USED ON A SYSTEM IN WHICH THE CONTENTS OF INTERNAL       00474300
C COMMON BLOCKS ARE NOT PRESERVED BETWEEN CALLS, THE USER SHOULD        00474400
C DECLARE THE ABOVE COMMON BLOCKS IN HIS MAIN PROGRAM TO INSURE         00474500
C THAT THEIR CONTENTS ARE PRESERVED.                                    00474600
C                                                                       00474700
C IF THE SOLUTION OF A GIVEN PROBLEM BY LSODAR IS TO BE INTERRUPTED     00474800
C AND THEN LATER CONTINUED, SUCH AS WHEN RESTARTING AN INTERRUPTED RUN  00474900
C OR ALTERNATING BETWEEN TWO OR MORE PROBLEMS, THE USER SHOULD SAVE,    00475000
C FOLLOWING THE RETURN FROM THE LAST LSODAR CALL PRIOR TO THE           00475100
C INTERRUPTION, THE CONTENTS OF THE CALL SEQUENCE VARIABLES AND THE     00475200
C INTERNAL COMMON BLOCKS, AND LATER RESTORE THESE VALUES BEFORE THE     00475300
C NEXT LSODAR CALL FOR THAT PROBLEM.  TO SAVE AND RESTORE THE COMMON    00475400
C BLOCKS, USE SUBROUTINES SVCAR AND RSCAR (SEE PART II ABOVE).          00475500
C                                                                       00475600
C-----------------------------------------------------------------------00475700
C PART IV.  OPTIONALLY REPLACEABLE SOLVER ROUTINES.                     00475800
C                                                                       00475900
C BELOW IS A DESCRIPTION OF A ROUTINE IN THE LSODAR PACKAGE WHICH       00476000
C RELATES TO THE MEASUREMENT OF ERRORS, AND CAN BE                      00476100
C REPLACED BY A USER-SUPPLIED VERSION, IF DESIRED.  HOWEVER, SINCE SUCH 00476200
C A REPLACEMENT MAY HAVE A MAJOR IMPACT ON PERFORMANCE, IT SHOULD BE    00476300
C DONE ONLY WHEN ABSOLUTELY NECESSARY, AND ONLY WITH GREAT CAUTION.     00476400
C (NOTE.. THE MEANS BY WHICH THE PACKAGE VERSION OF A ROUTINE IS        00476500
C SUPERSEDED BY THE USER-S VERSION MAY BE SYSTEM-DEPENDENT.)            00476600
C                                                                       00476700
C (A) EWSET.                                                            00476800
C THE FOLLOWING SUBROUTINE IS CALLED JUST BEFORE EACH INTERNAL          00476900
C INTEGRATION STEP, AND SETS THE ARRAY OF ERROR WEIGHTS, EWT, AS        00477000
C DESCRIBED UNDER ITOL/RTOL/ATOL ABOVE..                                00477100
C     SUBROUTINE EWSET (NEQ, ITOL, RTOL, ATOL, YCUR, EWT)               00477200
C WHERE NEQ, ITOL, RTOL, AND ATOL ARE AS IN THE LSODAR CALL SEQUENCE,   00477300
C YCUR CONTAINS THE CURRENT DEPENDENT VARIABLE VECTOR, AND              00477400
C EWT IS THE ARRAY OF WEIGHTS SET BY EWSET.                             00477500
C                                                                       00477600
C IF THE USER SUPPLIES THIS SUBROUTINE, IT MUST RETURN IN EWT(I)        00477700
C (I = 1,...,NEQ) A POSITIVE QUANTITY SUITABLE FOR COMPARING ERRORS     00477800
C IN Y(I) TO.  THE EWT ARRAY RETURNED BY EWSET IS PASSED TO THE         00477900
C VMNORM ROUTINE, AND ALSO USED BY LSODAR IN THE COMPUTATION            00478000
C OF THE OPTIONAL OUTPUT IMXER, AND THE INCREMENTS FOR DIFFERENCE       00478100
C QUOTIENT JACOBIANS.                                                   00478200
C                                                                       00478300
C IN THE USER-SUPPLIED VERSION OF EWSET, IT MAY BE DESIRABLE TO USE     00478400
C THE CURRENT VALUES OF DERIVATIVES OF Y.  DERIVATIVES UP TO ORDER NQ   00478500
C ARE AVAILABLE FROM THE HISTORY ARRAY YH, DESCRIBED ABOVE UNDER        00478600
C OPTIONAL OUTPUTS.  IN EWSET, YH IS IDENTICAL TO THE YCUR ARRAY,       00478700
C EXTENDED TO NQ + 1 COLUMNS WITH A COLUMN LENGTH OF NYH AND SCALE      00478800
C FACTORS OF H**J/FACTORIAL(J).  ON THE FIRST CALL FOR THE PROBLEM,     00478900
C GIVEN BY NST = 0, NQ IS 1 AND H IS TEMPORARILY SET TO 1.0.            00479000
C THE QUANTITIES NQ, NYH, H, AND NST CAN BE OBTAINED BY INCLUDING       00479100
C IN EWSET THE STATEMENTS..                                             00479200
C     DOUBLE PRECISION H, RLS                                           00479300
C     COMMON /LS0001/ RLS(219),ILS(39)                                  00479400
C     NQ = ILS(35)                                                      00479500
C     NYH = ILS(14)                                                     00479600
C     NST = ILS(36)                                                     00479700
C     H = RLS(213)                                                      00479800
C THUS, FOR EXAMPLE, THE CURRENT VALUE OF DY/DT CAN BE OBTAINED AS      00479900
C YCUR(NYH+I)/H  (I=1,...,NEQ)  (AND THE DIVISION BY H IS               00480000
C UNNECESSARY WHEN NST = 0).                                            00480100
C-----------------------------------------------------------------------00480200
C-----------------------------------------------------------------------00480300
C OTHER ROUTINES IN THE LSODAR PACKAGE.                                 00480400
```

```
C                                                                       00480500
C IN ADDITION TO SUBROUTINE LSODAR, THE LSODAR PACKAGE INCLUDES THE     00480600
C FOLLOWING SUBROUTINES AND FUNCTION ROUTINES..                         00480700
C  RCHEK     DOES PRELIMINARY CHECKING FOR ROOTS, AND SERVES AS AN      00480800
C            INTERFACE BETWEEN SUBROUTINE LSODAR AND SUBROUTINE ROOTS.  00480900
C  ROOTS     FINDS THE LEFTMOST ROOT OF A SET OF FUNCTIONS.             00481000
C  INTDY     COMPUTES AN INTERPOLATED VALUE OF THE Y VECTOR AT T = TOUT. 00481100
C  STODA     IS THE CORE INTEGRATOR, WHICH DOES ONE STEP OF THE         00481200
C            INTEGRATION AND THE ASSOCIATED ERROR CONTROL.              00481300
C  CFODE     SETS ALL METHOD COEFFICIENTS AND TEST CONSTANTS.           00481400
C  PRJA      COMPUTES AND PREPROCESSES THE JACOBIAN MATRIX J = DF/DY    00481500
C            AND THE NEWTON ITERATION MATRIX P = I - H*EL0*J.           00481600
C  SOLSY     MANAGES SOLUTION OF LINEAR SYSTEM IN CHORD ITERATION.      00481700
C  EWSET     SETS THE ERROR WEIGHT VECTOR EWT BEFORE EACH STEP.         00481800
C  VMNORM    COMPUTES THE WEIGHTED MAX-NORM OF A VECTOR.                00481900
C  FNORM     COMPUTES THE NORM OF A FULL MATRIX CONSISTENT WITH THE     00482000
C            WEIGHTED MAX-NORM ON VECTORS.                              00482100
C  BNORM     COMPUTES THE NORM OF A BAND MATRIX CONSISTENT WITH THE     00482200
C            WEIGHTED MAX-NORM ON VECTORS.                              00482300
C  SVCAR AND RSCAR   ARE USER-CALLABLE ROUTINES TO SAVE AND RESTORE,    00482400
C            RESPECTIVELY, THE CONTENTS OF THE INTERNAL COMMON BLOCKS.  00482500
C  DGEFA AND DGESL   ARE ROUTINES FROM LINPACK FOR SOLVING FULL         00482600
C            SYSTEMS OF LINEAR ALGEBRAIC EQUATIONS.                     00482700
C  DGBFA AND DGBSL   ARE ROUTINES FROM LINPACK FOR SOLVING BANDED       00482800
C            LINEAR SYSTEMS.                                            00482900
C  DAXPY, DSCAL, IDAMAX, DDOT, AND DCOPY   ARE BASIC LINEAR ALGEBRA     00483000
C            MODULES (BLAS) USED BY THE ABOVE LINPACK ROUTINES.         00483100
C  D1MACH    COMPUTES THE UNIT ROUNDOFF IN A MACHINE-INDEPENDENT MANNER. 00483200
C  XERRWV, XSETUN, AND XSETF   HANDLE THE PRINTING OF ALL ERROR         00483300
C            MESSAGES AND WARNINGS.  XERRWV IS MACHINE-DEPENDENT.       00483400
C NOTE..  VMNORM, FNORM, BNORM, IDAMAX, DDOT, AND D1MACH ARE FUNCTION   00483500
C ROUTINES.  ALL THE OTHERS ARE SUBROUTINES.                            00483600
C                                                                       00483700
C THE INTRINSIC AND EXTERNAL ROUTINES USED BY LSODAR ARE..              00483800
C DABS, DMAX1, DMIN1, DFLOAT, MAXO, MINO, MOD, DSIGN, DSQRT, AND WRITE. 00483900
C                                                                       00484000
C A BLOCK DATA SUBPROGRAM IS ALSO INCLUDED WITH THE PACKAGE,            00484100
C FOR LOADING SOME OF THE VARIABLES IN INTERNAL COMMON.                 00484200
C                                                                       00484300
C-----------------------------------------------------------------------00484400
C THE FOLLOWING CARD IS FOR OPTIMIZED COMPILATION ON LLL COMPILERS.     00484500
CLLL. OPTIMIZE                                                          00484600
C-----------------------------------------------------------------------00484700
      EXTERNAL PRJA, SOLSY                                              00484800
      INTEGER ILLIN, INIT, LYH, LEWT, LACOR, LSAVF, LWM, LIWM,          00484900
     1   MXSTEP, MXHNIL, NHNIL, NTREP, NSLAST, NYH, IOWNS              00485000
      INTEGER ICF, IERPJ, IERSL, JCUR, JSTART, KFLAG, L, METH, MITER,   00485100
     1   MAXORD, MAXCOR, MSBP, MXNCF, N, NQ, NST, NFE, NJE, NQU        00485200
      INTEGER INSUFR, INSUFI, IXPR, IOWNS2, JTYP, MUSED, MXORDN, MXORDS 00485300
      INTEGER LGO, LG1, LGX, IOWNR3, IRFND, ITASKC, NGC, NGE            00485400
      INTEGER I, I1, I2, IFLAG, IMXER, KGO, LFO,                        00485500
     1   LENIW, LENRW, LENWM, ML, MORD, MU, MXHNLO, MXSTPO              00485600
      INTEGER LEN1, LEN1C, LEN1N, LEN1S, LEN2, LENIWC,                  00485700
     1   LENRWC, LENRWN, LENRWS                                         00485800
      INTEGER IRFP, IRT, LENYH, LYHNEW                                  00485900
      DOUBLE PRECISION TRET, ROWNS,                                     00486000
     1   CCMAX, ELO, H, HMIN, HMXI, HU, RC, TN, UROUND                  00486100
      DOUBLE PRECISION TSW, ROWNS2, PDNORM                              00486200
      DOUBLE PRECISION ROWNR3, TO, TLAST, TOUTC                         00486300
      DOUBLE PRECISION ATOLI, AYI, BIG, EWTI, HO, HMAX, HMX, RH, RTOLI, 00486400
     1   TCRIT, TDIST, TNEXT, TOL, TOLSF, TP, SIZE, SUM, WO,            00486500
     2   D1MACH, VMNORM                                                 00486600
      DIMENSION MORD(2)                                                 00486700
      LOGICAL IHIT                                                      00486800
C-----------------------------------------------------------------------00486900
C THE FOLLOWING THREE INTERNAL COMMON BLOCKS CONTAIN                    00487000
C (A) VARIABLES WHICH ARE LOCAL TO ANY SUBROUTINE BUT WHOSE VALUES MUST 00487100
C     BE PRESERVED BETWEEN CALLS TO THE ROUTINE (OWN VARIABLES), AND    00487200
C (B) VARIABLES WHICH ARE COMMUNICATED BETWEEN SUBROUTINES.             00487300
C THE STRUCTURE OF EACH BLOCK IS AS FOLLOWS..  ALL REAL VARIABLES ARE   00487400
C LISTED FIRST, FOLLOWED BY ALL INTEGERS.  WITHIN EACH TYPE, THE        00487500
C VARIABLES ARE GROUPED WITH THOSE LOCAL TO SUBROUTINE LSODAR FIRST,    00487600
```

```
C THEN THOSE LOCAL TO SUBROUTINE ROOTS OR SUBROUTINE STODA                00487700
C (NO OTHER ROUTINES HAVE OWN VARIABLES), AND FINALLY THOSE USED          00487800
C FOR COMMUNICATION.  THE BLOCK LS0001 IS DECLARED IN SUBROUTINES         00487900
C LSODAR, INTDY, STODA, PRJA, AND SOLSY.  THE BLOCK LSA001 IS DECLARED    00488000
C IN SUBROUTINES LSODAR, STODA, AND PRJA.  THE BLOCK LSR001 IS DECLARED   00488100
C IN SUBROUTINES LSODAR, RCHEK, AND ROOTS.  GROUPS OF VARIABLES ARE       00488200
C REPLACED BY DUMMY ARRAYS IN THE COMMON DECLARATIONS IN ROUTINES         00488300
C WHERE THOSE VARIABLES ARE NOT USED.                                     00488400
C-----------------------------------------------------------------------00488500
      COMMON /LS0001/ TRET, ROWNS(209),                                   00488600
     1   CCMAX, ELO, H, HMIN, HMXI, HU, RC, TN, UROUND,                   00488700
     2   ILLIN, INIT, LYH, LEWT, LACOR, LSAVF, LWM, LIWM,                 00488800
     3   MXSTEP, MXHNIL, NHNIL, NTREP, NSLAST, NYH, IOWNS(6),             00488900
     4   ICF, IERPJ, IERSL, JCUR, JSTART, KFLAG, L, METH, MITER,          00489000
     5   MAXORD, MAXCOR, MSBP, MXNCF, N, NQ, NST, NFE, NJE, NQU           00489100
      COMMON /LSA001/ TSW, ROWNS2(20), PDNORM,                            00489200
     1   INSUFR, INSUFI, IXPR, IOWNS2(2), JTYP, MUSED, MXORDN, MXORDS     00489300
      COMMON /LSR001/ ROWNR3(2), TO, TLAST, TOUTC,                        00489400
     1   LGO, LG1, LGX, IOWNR3(2), IRFND, ITASKC, NGC, NGE                00489500
C                                                                         00489600
      DATA MORD(1),MORD(2)/12,5/, MXSTPO/500/, MXHNLO/10/                 00489700
C-----------------------------------------------------------------------00489800
C BLOCK A.                                                                00489900
C THIS CODE BLOCK IS EXECUTED ON EVERY CALL.                              00490000
C IT TESTS ISTATE AND ITASK FOR LEGALITY AND BRANCHES APPROPIATELY.       00490100
C IF ISTATE .GT. 1 BUT THE FLAG INIT SHOWS THAT INITIALIZATION HAS        00490200
C NOT YET BEEN DONE, AN ERROR RETURN OCCURS.                              00490300
C IF ISTATE = 1 AND TOUT = T, JUMP TO BLOCK G AND RETURN IMMEDIATELY.     00490400
C-----------------------------------------------------------------------00490500
      IF (ISTATE .LT. 1 .OR. ISTATE .GT. 3) GO TO 601                     00490700
      IF (ITASK .LT. 1 .OR. ITASK .GT. 5) GO TO 602                       00490800
      ITASKC = ITASK                                                      00490900
      IF (ISTATE .EQ. 1) GO TO 10                                         00491000
      IF (INIT .EQ. 0) GO TO 603                                          00491100
      IF (ISTATE .EQ. 2) GO TO 200                                        00491200
      GO TO 20                                                            00491300
 10   INIT = 0                                                            00491400
      IF (TOUT .EQ. T) GO TO 430                                          00491500
 20   NTREP = 0                                                           00491600
C-----------------------------------------------------------------------00491700
C BLOCK B.                                                                00491800
C THE NEXT CODE BLOCK IS EXECUTED FOR THE INITIAL CALL (ISTATE = 1),      00491900
C OR FOR A CONTINUATION CALL WITH PARAMETER CHANGES (ISTATE = 3).         00492000
C IT CONTAINS CHECKING OF ALL INPUTS AND VARIOUS INITIALIZATIONS.         00492100
C                                                                         00492200
C FIRST CHECK LEGALITY OF THE NON-OPTIONAL INPUTS NEQ, ITOL, IOPT,        00492300
C JT, ML, MU, AND NG.                                                     00492400
C-----------------------------------------------------------------------00492500
      IF (NEQ .LE. 0) GO TO 604                                           00492600
      IF (ISTATE .EQ. 1) GO TO 25                                         00492700
      IF (NEQ .GT. N) GO TO 605                                           00492800
 25   N = NEQ                                                             00492900
      IF (ITOL .LT. 1 .OR. ITOL .GT. 4) GO TO 606                         00493000
      IF (IOPT .LT. 0 .OR. IOPT .GT. 1) GO TO 607                         00493100
      IF (JT .EQ. 3 .OR. JT .LT. 1 .OR. JT .GT. 5) GO TO 608              00493200
      JTYP = JT                                                           00493300
      IF (JT .LE. 2) GO TO 30                                             00493400
      ML = IWORK(1)                                                       00493500
      MU = IWORK(2)                                                       00493600
      IF (ML .LT. 0 .OR. ML .GE. N) GO TO 609                             00493700
      IF (MU .LT. 0 .OR. MU .GE. N) GO TO 610                             00493800
 30   CONTINUE                                                            00493900
      IF (NG .LT. 0) GO TO 630                                            00494000
      IF (ISTATE .EQ. 1) GO TO 35                                         00494100
      IF (IRFND .EQ. 0 .AND. NG .NE. NGC) GO TO 631                       00494200
 35   NGC = NG                                                            00494300
C NEXT PROCESS AND CHECK THE OPTIONAL INPUTS. -------------------------00494400
      IF (IOPT .EQ. 1) GO TO 40                                           00494500
      IXPR = 0                                                            00494600
      MXSTEP = MXSTPO                                                     00494700
      MXHNIL = MXHNLO                                                     00494800
      HMXI = 0.0D0                                                        00494900
```

```
      HMIN = O.ODO                                                      00495000
      IF (ISTATE .NE. 1) GO TO 60                                       00495100
      HO = O.ODO                                                        00495200
      MXORDN = MORD(1)                                                  00495300
      MXORDS = MORD(2)                                                  00495400
      GO TO 60                                                          00495500
 40   IXPR = IWORK(5)                                                   00495600
      IF (IXPR .LT. O .OR. IXPR .GT. 1) GO TO 611                       00495700
      MXSTEP = IWORK(6)                                                 00495800
      IF (MXSTEP .LT. O) GO TO 612                                      00495900
      IF (MXSTEP .EQ. O) MXSTEP = MXSTPO                                00496000
      MXHNIL = IWORK(7)                                                 00496100
      IF (MXHNIL .LT. O) GO TO 613                                      00496200
      IF (MXHNIL .EQ. O) MXHNIL = MXHNLO                                00496300
      IF (ISTATE .NE. 1) GO TO 50                                       00496400
      HO = RWORK(5)                                                     00496500
      MXORDN = IWORK(8)                                                 00496600
      IF (MXORDN .LT. O) GO TO 628                                      00496700
      IF (MXORDN .EQ. O) MXORDN = 100                                   00496800
      MXORDN = MINO(MXORDN,MORD(1))                                     00496900
      MXORDS = IWORK(9)                                                 00497000
      IF (MXORDS .LT. O) GO TO 629                                      00497100
      IF (MXORDS .EQ. O) MXORDS = 100                                   00497200
      MXORDS = MINO(MXORDS,MORD(2))                                     00497300
      IF ((TOUT - T)*HO .LT. O.ODO) GO TO 614                           00497400
 50   HMAX = RWORK(6)                                                   00497500
      IF (HMAX .LT. O.ODO) GO TO 615                                    00497600
      HMXI = O.ODO                                                      00497700
      IF (HMAX .GT. O.ODO) HMXI = 1.ODO/HMAX                            00497800
      HMIN = RWORK(7)                                                   00497900
      IF (HMIN .LT. O.ODO) GO TO 616                                    00498000
C-----------------------------------------------------------------------00498100
C SET WORK ARRAY POINTERS AND CHECK LENGTHS LRW AND LIW.                00498200
C IF ISTATE = 1, METH IS INITIALIZED TO 1 HERE TO FACILITATE THE        00498300
C CHECKING OF WORK SPACE LENGTHS.                                       00498400
C POINTERS TO SEGMENTS OF RWORK AND IWORK ARE NAMED BY PREFIXING L TO   00498500
C THE NAME OF THE SEGMENT.  E.G., THE SEGMENT YH STARTS AT RWORK(LYH).  00498600
C SEGMENTS OF RWORK (IN ORDER) ARE DENOTED  GO, G1, GX, YH, WM,         00498700
C EWT, SAVF, ACOR.                                                      00498800
C IF THE LENGTHS PROVIDED ARE INSUFFICIENT FOR THE CURRENT METHOD,      00498900
C AN ERROR RETURN OCCURS.  THIS IS TREATED AS ILLEGAL INPUT ON THE      00499000
C FIRST CALL, BUT AS A PROBLEM INTERRUPTION WITH ISTATE = -7 ON A       00499100
C CONTINUATION CALL.  IF THE LENGTHS ARE SUFFICIENT FOR THE CURRENT     00499200
C METHOD BUT NOT FOR BOTH METHODS, A WARNING MESSAGE IS SENT.           00499300
C-----------------------------------------------------------------------00499400
 60   IF (ISTATE .EQ. 1) METH = 1                                       00499500
      IF (ISTATE .EQ. 1) NYH = N                                        00499600
      LGO = 21                                                          00499700
      LG1 = LGO + NG                                                    00499800
      LGX = LG1 + NG                                                    00499900
      LYHNEW = LGX + NG                                                 00500000
      IF (ISTATE .EQ. 1) LYH = LYHNEW                                   00500100
      IF (LYHNEW .EQ. LYH) GO TO 62                                     00500200
C IF ISTATE = 3 AND NG WAS CHANGED, SHIFT YH TO ITS NEW LOCATION. ------00500300
      LENYH = L*NYH                                                     00500400
      IF (LRW .LT. LYHNEW-1+LENYH) GO TO 62                             00500500
      I1 = 1                                                            00500600
      IF (LYHNEW .GT. LYH) I1 = -1                                      00500700
      CALL DCOPY (LENYH, RWORK(LYH), I1, RWORK(LYHNEW), I1)             00500800
      LYH = LYHNEW                                                      00500900
 62   CONTINUE                                                          00501000
      LEN1N = LYHNEW - 1 + (MXORDN + 1)*NYH                             00501100
      LEN1S = LYHNEW - 1 + (MXORDS + 1)*NYH                             00501200
      LWM = LEN1S + 1                                                   00501300
      IF (JT .LE. 2) LENWM = N*N + 2                                    00501400
      IF (JT .GE. 4) LENWM = (2*ML + MU + 1)*N + 2                      00501500
      LEN1S = LEN1S + LENWM                                             00501600
      LEN1C = LEN1N                                                     00501700
      IF (METH .EQ. 2) LEN1C = LEN1S                                    00501800
      LEN1 = MAXO(LEN1N,LEN1S)                                          00501900
      LEN2 = 3*N                                                        00502000
      LENRW = LEN1 + LEN2                                               00502100
```

```
      LENRWN = LEN1N + LEN2                                              00502200
      LENRWS = LEN1S + LEN2                                              00502300
      LENRWC = LEN1C + LEN2                                              00502400
      IWORK(17) = LENRW                                                  00502500
      LIWM = 1                                                           00502600
      LENIW = 20 + N                                                     00502700
      LENIWC = 20                                                        00502800
      IF (METH .EQ. 2) LENIWC = LENIW                                    00502900
      IWORK(18) = LENIW                                                  00503000
C-----------------------------------------------------------            00503100
      IF (ISTATE .EQ. 1 .AND. LRW .LT. LENRWC) GO TO 617                 00503200
      IF (ISTATE .EQ. 1 .AND. LIW .LT. LENIWC) GO TO 618                 00503300
      IF (ISTATE .EQ. 3 .AND. LRW .LT. LENRWC) GO TO 550                 00503400
      IF (ISTATE .EQ. 3 .AND. LIW .LT. LENIWC) GO TO 555                 00503500
      LEWT = LEN1 + 1                                                    00503600
      INSUFR = 0                                                         00503700
      IF (LRW .GE. LENRW) GO TO 65                                       00503800
      INSUFR = 2                                                         00503900
      LEWT = LEN1C + 1                                                   00504000
      CALL XERRWV(                                                       00504100
     1   1, 103, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                           00504200
      CALL XERRWV(                                                       00504600
     1   1, 103, 1, 2, LENRW, LRW, 0, 0.0D0, 0.0D0)                     00504700
 65   LSAVF = LEWT + N                                                   00504800
      LACOR = LSAVF + N                                                  00504900
      INSUFI = 0                                                         00505000
      IF (LIW .GE. LENIW) GO TO 70                                       00505100
      INSUFI = 2                                                         00505200
      CALL XERRWV(                                                       00505300
     1   2, 104, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                           00505400
C     CALL XERRWV(                                                       00505500
C    1  60H      MAY NOT BE LATER.  INTEGRATION WILL PROCEED ANYWAY.    ,00505600
C    1   60, 104, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                          00505700
      CALL XERRWV(                                                       00505800
     1   2, 104, 1, 2, LENIW, LIW, 0, 0.0D0, 0.0D0)                     00505900
 70   CONTINUE                                                           00506000
C CHECK RTOL AND ATOL FOR LEGALITY. ------------------------------------00506100
      RTOLI = RTOL(1)                                                    00506200
      ATOLI = ATOL(1)                                                    00506300
      DO 75 I = 1,N                                                      00506400
        IF (ITOL .GE. 3) RTOLI = RTOL(I)                                 00506500
        IF (ITOL .EQ. 2 .OR. ITOL .EQ. 4) ATOLI = ATOL(I)                00506600
        IF (RTOLI .LT. 0.0D0) GO TO 619                                  00506700
        IF (ATOLI .LT. 0.0D0) GO TO 620                                  00506800
 75     CONTINUE                                                         00506900
      IF (ISTATE .EQ. 1) GO TO 100                                       00507000
C IF ISTATE = 3, SET FLAG TO SIGNAL PARAMETER CHANGES TO STODA. --------00507100
      JSTART = -1                                                        00507200
      IF (N .EQ. NYH) GO TO 200                                          00507300
C NEQ WAS REDUCED.  ZERO PART OF YH TO AVOID UNDEFINED REFERENCES. -----00507400
      I1 = LYH + L*NYH                                                   00507500
      I2 = LYH + (MAXORD + 1)*NYH - 1                                    00507600
      IF (I1 .GT. I2) GO TO 200                                          00507700
      DO 95 I = I1,I2                                                    00507800
 95     RWORK(I) = 0.0D0                                                 00507900
      GO TO 200                                                          00508000
C------------------------------------------------------------------------00508100
C BLOCK C.                                                               00508200
C THE NEXT BLOCK IS FOR THE INITIAL CALL ONLY (ISTATE = 1).              00508300
C IT CONTAINS ALL REMAINING INITIALIZATIONS, THE INITIAL CALL TO F,      00508400
C AND THE CALCULATION OF THE INITIAL STEP SIZE.                          00508500
C THE ERROR WEIGHTS IN EWT ARE INVERTED AFTER BEING LOADED.              00508600
C------------------------------------------------------------------------00508700
 100  UROUND = D1MACH(4)                                                 00508800
      TN = T                                                             00508900
      TSW = T                                                            00509000
      MAXORD = MXORDN                                                    00509100
      IF (ITASK .NE. 4 .AND. ITASK .NE. 5) GO TO 110                     00509200
      TCRIT = RWORK(1)                                                   00509300
      IF ((TCRIT - TOUT)*(TOUT - T) .LT. 0.0D0) GO TO 625                00509400
      IF (H0 .NE. 0.0D0 .AND. (T + H0 - TCRIT)*H0 .GT. 0.0D0)            00509500
     1   H0 = TCRIT - T                                                  00509600
```

```
 110  JSTART = 0                                                        00509700
      NHNIL = 0                                                         00509800
      NST = 0                                                           00509900
      NJE = 0                                                           00510000
      NSLAST = 0                                                        00510100
      HU = 0.0D0                                                        00510200
      NQU = 0                                                           00510300
      MUSED = 0                                                         00510400
      MITER = 0                                                         00510500
      CCMAX = 0.3D0                                                     00510600
      MAXCOR = 3                                                        00510700
      MSBP = 20                                                         00510800
      MXNCF = 10                                                        00510900
C INITIAL CALL TO F.  (LFO POINTS TO YH(*,2).) -------------------------00511000
      LFO = LYH + NYH                                                   00511100
      CALL FEX (NEQ, T, Y, RWORK(LFO),INDX)                             00511200
      NFE = 1                                                           00511500
C LOAD THE INITIAL VALUE VECTOR IN YH. ---------------------------------00511600
      DO 115 I = 1,N                                                    00511700
 115    RWORK(I+LYH-1) = Y(I)                                           00511800
C LOAD AND INVERT THE EWT ARRAY.  (H IS TEMPORARILY SET TO 1.0.) -------00511900
      NQ = 1                                                            00512000
      H = 1.0D0                                                         00512100
      CALL EWSET (N, ITOL, RTOL, ATOL, RWORK(LYH), RWORK(LEWT))         00512200
      DO 120 I = 1,N                                                    00512300
        IF (RWORK(I+LEWT-1) .LE. 0.0D0) GO TO 621                       00512400
 120    RWORK(I+LEWT-1) = 1.0D0/RWORK(I+LEWT-1)                         00512500
C-----------------------------------------------------------------------00512600
C THE CODING BELOW COMPUTES THE STEP SIZE, H0, TO BE ATTEMPTED ON THE   00512700
C FIRST STEP, UNLESS THE USER HAS SUPPLIED A VALUE FOR THIS.            00512800
C FIRST CHECK THAT TOUT - T DIFFERS SIGNIFICANTLY FROM ZERO.            00512900
C A SCALAR TOLERANCE QUANTITY TOL IS COMPUTED, AS MAX(RTOL(I))          00513000
C IF THIS IS POSITIVE, OR MAX(ATOL(I)/ABS(Y(I))) OTHERWISE, ADJUSTED    00513100
C SO AS TO BE BETWEEN 100*UROUND AND 1.0E-3.                            00513200
C THEN THE COMPUTED VALUE H0 IS GIVEN BY..                              00513300
C                                                                       00513400
C   H0**(-2)  =  1./(TOL * W0**2)  +  TOL * (NORM(F))**2                00513500
C                                                                       00513600
C WHERE   W0     = MAX ( ABS(T), ABS(TOUT) ),                           00513700
C         F      = THE INITIAL VALUE OF THE VECTOR F(T,Y), AND          00513800
C         NORM() = THE WEIGHTED VECTOR NORM USED THROUGHOUT, GIVEN BY   00513900
C                  THE VMNORM FUNCTION ROUTINE, AND WEIGHTED BY THE     00514000
C                  TOLERANCES INITIALLY LOADED INTO THE EWT ARRAY.      00514100
C THE SIGN OF H0 IS INFERRED FROM THE INITIAL VALUES OF TOUT AND T.      00514200
C ABS(H0) IS MADE .LE. ABS(TOUT-T) IN ANY CASE.                         00514300
C-----------------------------------------------------------------------00514400
      IF (H0 .NE. 0.0D0) GO TO 180                                      00514500
      TDIST = DABS(TOUT - T)                                            00514600
      W0 = DMAX1(DABS(T),DABS(TOUT))                                    00514700
      IF (TDIST .LT. 2.0D0*UROUND*W0) GO TO 622                         00514800
      TOL = RTOL(1)                                                     00514900
      IF (ITOL .LE. 2) GO TO 140                                        00515000
      DO 130 I = 1,N                                                    00515100
 130    TOL = DMAX1(TOL,RTOL(I))                                        00515200
 140  IF (TOL .GT. 0.0D0) GO TO 160                                     00515300
      ATOLI = ATOL(1)                                                   00515400
      DO 150 I = 1,N                                                    00515500
        IF (ITOL .EQ. 2 .OR. ITOL .EQ. 4) ATOLI = ATOL(I)               00515600
        AYI = DABS(Y(I))                                                00515700
        IF (AYI .NE. 0.0D0) TOL = DMAX1(TOL,ATOLI/AYI)                  00515800
 150    CONTINUE                                                        00515900
 160  TOL = DMAX1(TOL,100.0D0*UROUND)                                   00516000
      TOL = DMIN1(TOL,0.001D0)                                          00516100
      SUM = VMNORM (N, RWORK(LFO), RWORK(LEWT))                         00516200
      SUM = 1.0D0/(TOL*W0*W0) + TOL*SUM**2                              00516300
      H0 = 1.0D0/DSQRT(SUM)                                             00516400
      H0 = DMIN1(H0,TDIST)                                              00516500
      H0 = DSIGN(H0,TOUT-T)                                             00516600
C ADJUST H0 IF NECESSARY TO MEET HMAX BOUND. ---------------------------00516700
 180  RH = DABS(H0)*HMXI                                                00516800
      IF (RH .GT. 1.0D0) H0 = H0/RH                                     00516900
C LOAD H WITH H0 AND SCALE YH(*,2) BY H0. ------------------------------00517000
```

```
      H = HO                                                            00517100
      DO 190 I = 1,N                                                    00517200
 190    RWORK(I+LFO-1) = HO*RWORK(I+LFO-1)                             00517300
C                                                                       00517400
C CHECK FOR A ZERO OF G AT T. ------------------------------------------00517500
      IRFND = 0                                                         00517600
      TOUTC = TOUT                                                      00517700
      IF (NGC .EQ. 0) GO TO 270                                         00517800
      CALL RCHEK (1, G, NEQ, Y, RWORK(LYH), NYH,                        00517900
     1   RWORK(LGO), RWORK(LG1), RWORK(LGX), JROOT, IRT,INDX)           00518000
      IF (IRT .EQ. 0) GO TO 270                                         00518100
      GO TO 632                                                         00518200
C-----------------------------------------------------------------------00518300
C BLOCK D.                                                              00518400
C THE NEXT CODE BLOCK IS FOR CONTINUATION CALLS ONLY (ISTATE = 2 OR 3)  00518500
C AND IS TO CHECK STOP CONDITIONS BEFORE TAKING A STEP.                 00518600
C FIRST, RCHEK IS CALLED TO CHECK FOR A ROOT WITHIN THE LAST STEP       00518700
C TAKEN, OTHER THAN THE LAST ROOT FOUND THERE, IF ANY.                  00518800
C IF ITASK = 2 OR 5, AND Y(TN) HAS NOT YET BEEN RETURNED TO THE USER    00518900
C BECAUSE OF AN INTERVENING ROOT, RETURN THROUGH BLOCK G.               00519000
C-----------------------------------------------------------------------00519100
 200  NSLAST = NST                                                      00519200
C                                                                       00519300
      IRFP = IRFND                                                      00519400
      IF (NGC .EQ. 0) GO TO 205                                         00519500
      IF (ITASK .EQ. 1 .OR. ITASK .EQ. 4) TOUTC = TOUT                  00519600
      CALL RCHEK (2, G, NEQ, Y, RWORK(LYH), NYH,                        00519700
     1   RWORK(LGO), RWORK(LG1), RWORK(LGX), JROOT, IRT,INDX)           00519800
      IF (IRT .NE. 1) GO TO 205                                         00519900
      IRFND = 1                                                         00520000
      ISTATE = 3                                                        00520100
      T = TO                                                            00520200
      GO TO 425                                                         00520300
 205  CONTINUE                                                          00520400
      IRFND = 0                                                         00520500
      IF (IRFP .EQ. 1 .AND. TLAST .NE. TN .AND. ITASK .EQ. 2) GO TO 400 00520600
C                                                                       00520700
      GO TO (210, 250, 220, 230, 240), ITASK                            00520800
 210  IF ((TN - TOUT)*H .LT. 0.0D0) GO TO 250                           00520900
      CALL INTDY (TOUT, 0, RWORK(LYH), NYH, Y, IFLAG)                   00521000
      IF (IFLAG .NE. 0) GO TO 627                                       00521100
      T = TOUT                                                          00521200
      GO TO 420                                                         00521300
 220  TP = TN - HU*(1.0D0 + 100.0D0*UROUND)                             00521400
      IF ((TP - TOUT)*H .GT. 0.0D0) GO TO 623                           00521500
      IF ((TN - TOUT)*H .LT. 0.0D0) GO TO 250                           00521600
      T = TN                                                            00521700
      GO TO 400                                                         00521800
 230  TCRIT = RWORK(1)                                                  00521900
      IF ((TN - TCRIT)*H .GT. 0.0D0) GO TO 624                          00522000
      IF ((TCRIT - TOUT)*H .LT. 0.0D0) GO TO 625                        00522100
      IF ((TN - TOUT)*H .LT. 0.0D0) GO TO 245                           00522200
      CALL INTDY (TOUT, 0, RWORK(LYH), NYH, Y, IFLAG)                   00522300
      IF (IFLAG .NE. 0) GO TO 627                                       00522400
      T = TOUT                                                          00522500
      GO TO 420                                                         00522600
 240  TCRIT = RWORK(1)                                                  00522700
      IF ((TN - TCRIT)*H .GT. 0.0D0) GO TO 624                          00522800
 245  HMX = DABS(TN) + DABS(H)                                          00522900
      IHIT = DABS(TN - TCRIT) .LE. 100.0D0*UROUND*HMX                   00523000
      IF (IHIT) T = TCRIT                                               00523100
      IF (IRFP .EQ. 1 .AND. TLAST .NE. TN .AND. ITASK .EQ. 5) GO TO 400 00523200
      IF (IHIT) GO TO 400                                               00523300
      TNEXT = TN + H*(1.0D0 + 4.0D0*UROUND)                             00523400
      IF ((TNEXT - TCRIT)*H .LE. 0.0D0) GO TO 250                       00523500
      H = (TCRIT - TN)*(1.0D0 - 4.0D0*UROUND)                           00523600
      IF (ISTATE .EQ. 2) JSTART = -2                                    00523700
C-----------------------------------------------------------------------00523800
C BLOCK E.                                                              00523900
C THE NEXT BLOCK IS NORMALLY EXECUTED FOR ALL CALLS AND CONTAINS        00524000
C THE CALL TO THE ONE-STEP CORE INTEGRATOR STODA.                       00524100
C                                                                       00524200
```

```
C THIS IS A LOOPING POINT FOR THE INTEGRATION STEPS.                    00524300
C                                                                       00524400
C FIRST CHECK FOR TOO MANY STEPS BEING TAKEN, UPDATE EWT (IF NOT AT     00524500
C START OF PROBLEM), CHECK FOR TOO MUCH ACCURACY BEING REQUESTED, AND   00524600
C CHECK FOR H BELOW THE ROUNDOFF LEVEL IN T.                            00524700
C-----------------------------------------------------------------------00524800
 250  CONTINUE                                                          00524900
      IF (METH .EQ. MUSED) GO TO 255                                    00525000
      IF (INSUFR .EQ. 1) GO TO 550                                      00525100
      IF (INSUFI .EQ. 1) GO TO 555                                      00525200
 255  IF ((NST-NSLAST) .GE. MXSTEP) GO TO 500                           00525300
      CALL EWSET (N, ITOL, RTOL, ATOL, RWORK(LYH), RWORK(LEWT))         00525400
      DO 260 I = 1,N                                                    00525500
        IF (RWORK(I+LEWT-1) .LE. 0.0D0) GO TO 510                       00525600
 260    RWORK(I+LEWT-1) = 1.0D0/RWORK(I+LEWT-1)                         00525700
 270  TOLSF = UROUND*VMNORM (N, RWORK(LYH), RWORK(LEWT))                00525800
      IF (TOLSF .LE. 0.01D0) GO TO 280                                  00525900
      TOLSF = TOLSF*200.0D0                                             00526000
      IF (NST .EQ. 0) GO TO 626                                         00526100
      GO TO 520                                                         00526200
 280  IF ((TN + H) .NE. TN) GO TO 290                                   00526300
      NHNIL = NHNIL + 1                                                 00526400
      IF (NHNIL .GT. MXHNIL) GO TO 290                                  00526500
      CALL XERRWV(                                                      00526600
     1   3, 101, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                           00526700
C     CALL XERRWV(                                                      00526800
C    1  60H      SUCH THAT IN THE MACHINE, T + H = T ON THE NEXT STEP  ,00526900
C    1   60, 101, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                         00527000
      CALL XERRWV(                                                      00527100
     1   3, 101, 1, 0, 0, 0, 2, TN, H)                                  00527200
      IF (NHNIL .LT. MXHNIL) GO TO 290                                  00527300
      CALL XERRWV(                                                      00527400
     1   4, 102, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                           00527500
      CALL XERRWV(                                                      00527600
     1   4, 102, 1, 1, MXHNIL, 0, 0, 0.0D0, 0.0D0)                      00527700
 290  CONTINUE                                                          00527800
C-----------------------------------------------------------------------00527900
C     CALL STODA(NEQ,Y,YH,NYH,YH,EWT,SAVF,ACOR,WM,IWM,F,JAC,PJAC,SOLSY) 00528000
C-----------------------------------------------------------------------00528100
      CALL STODA (NEQ, Y, RWORK(LYH), NYH, RWORK(LYH), RWORK(LEWT),     00528200
     1   RWORK(LSAVF), RWORK(LACOR), RWORK(LWM), IWORK(LIWM),           00528300
     2   F, JAC, PJAC, SOLSY,INDX)                                      00528400
      KGO = 1 - KFLAG                                                   00528500
      GO TO (300, 530, 540), KGO                                        00528600
C-----------------------------------------------------------------------00528700
C BLOCK F.                                                              00528800
C THE FOLLOWING BLOCK HANDLES THE CASE OF A SUCCESSFUL RETURN FROM THE  00528900
C CORE INTEGRATOR (KFLAG = 0).                                          00529000
C IF A METHOD SWITCH WAS JUST MADE, RECORD TSW, RESET MAXORD,           00529100
C SET JSTART TO -1 TO SIGNAL STODA TO COMPLETE THE SWITCH,              00529200
C AND DO EXTRA PRINTING OF DATA IF IXPR = 1.                            00529300
C THEN CALL RCHEK TO CHECK FOR A ROOT WITHIN THE LAST STEP.             00529400
C THEN, IF NO ROOT WAS FOUND, CHECK FOR STOP CONDITIONS.                00529500
C-----------------------------------------------------------------------00529600
 300  INIT = 1                                                          00529700
      IF (METH .EQ. MUSED) GO TO 310                                    00529800
      TSW = TN                                                          00529900
      MAXORD = MXORDN                                                   00530000
      IF (METH .EQ. 2) MAXORD = MXORDS                                  00530100
      IF (METH .EQ. 2) RWORK(LWM) = DSQRT(UROUND)                       00530200
      INSUFR = MINO(INSUFR,1)                                           00530300
      INSUFI = MINO(INSUFI,1)                                           00530400
      JSTART = -1                                                       00530500
      IF (IXPR .EQ. 0) GO TO 310                                        00530600
      IF (METH .EQ. 2) CALL XERRWV(                                     00530700
     1   5, 105, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                           00530800
      IF (METH .EQ. 1) CALL XERRWV(                                     00530900
     1   6, 106, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                           00531000
      CALL XERRWV(                                                      00531100
     1   7, 107, 1, 1, NST, 0, 2, TN, H)                                00531200
 310  CONTINUE                                                          00531300
C                                                                       00531400
```

```
      IF (NGC .EQ. 0) GO TO 315                                          00531500
      CALL RCHEK (3, G, NEQ, Y, RWORK(LYH), NYH,                         00531600
     1   RWORK(LG0), RWORK(LG1), RWORK(LGX), JROOT, IRT,INDX)            00531700
      IF (IRT .NE. 1) GO TO 315                                          00531800
      IRFND = 1                                                          00531900
      ISTATE = 3                                                         00532000
      T = TO                                                             00532100
      GO TO 425                                                          00532200
 315  CONTINUE                                                           00532300
C                                                                        00532400
      GO TO (320, 400, 330, 340, 350), ITASK                             00532500
C ITASK = 1.  IF TOUT HAS BEEN REACHED, INTERPOLATE. -------------------00532600
 320  IF ((TN - TOUT)*H .LT. 0.0D0) GO TO 250                            00532700
      CALL INTDY (TOUT, 0, RWORK(LYH), NYH, Y, IFLAG)                    00532800
      T = TOUT                                                           00532900
      GO TO 420                                                          00533000
C ITASK = 3.  JUMP TO EXIT IF TOUT WAS REACHED. ------------------------00533100
 330  IF ((TN - TOUT)*H .GE. 0.0D0) GO TO 400                            00533200
      GO TO 250                                                          00533300
C ITASK = 4.  SEE IF TOUT OR TCRIT WAS REACHED.  ADJUST H IF NECESSARY. 00533400
 340  IF ((TN - TOUT)*H .LT. 0.0D0) GO TO 345                            00533500
      CALL INTDY (TOUT, 0, RWORK(LYH), NYH, Y, IFLAG)                    00533600
      T = TOUT                                                           00533700
      GO TO 420                                                          00533800
 345  HMX = DABS(TN) + DABS(H)                                           00533900
      IHIT = DABS(TN - TCRIT) .LE. 100.0D0*UROUND*HMX                    00534000
      IF (IHIT) GO TO 400                                                00534100
      TNEXT = TN + H*(1.0D0 + 4.0D0*UROUND)                              00534200
      IF ((TNEXT - TCRIT)*H .LE. 0.0D0) GO TO 250                        00534300
      H = (TCRIT - TN)*(1.0D0 - 4.0D0*UROUND)                            00534400
      JSTART = -2                                                        00534500
      GO TO 250                                                          00534600
C ITASK = 5.  SEE IF TCRIT WAS REACHED AND JUMP TO EXIT. ---------------00534700
 350  HMX = DABS(TN) + DABS(H)                                           00534800
      IHIT = DABS(TN - TCRIT) .LE. 100.0D0*UROUND*HMX                    00534900
C------------------------------------------------------------------------00535000
C BLOCK G.                                                               00535100
C THE FOLLOWING BLOCK HANDLES ALL SUCCESSFUL RETURNS FROM LSODAR.        00535200
C IF ITASK .NE. 1, Y IS LOADED FROM YH AND T IS SET ACCORDINGLY.         00535300
C ISTATE IS SET TO 2, THE ILLEGAL INPUT COUNTER IS ZEROED, AND THE       00535400
C OPTIONAL OUTPUTS ARE LOADED INTO THE WORK ARRAYS BEFORE RETURNING.     00535500
C IF ISTATE = 1 AND TOUT = T, THERE IS A RETURN WITH NO ACTION TAKEN,    00535600
C EXCEPT THAT IF THIS HAS HAPPENED REPEATEDLY, THE RUN IS TERMINATED.    00535700
C------------------------------------------------------------------------00535800
 400  DO 410 I = 1,N                                                     00535900
 410    Y(I) = RWORK(I+LYH-1)                                            00536000
      T = TN                                                             00536100
      IF (ITASK .NE. 4 .AND. ITASK .NE. 5) GO TO 420                     00536200
      IF (IHIT) T = TCRIT                                                00536300
 420  ISTATE = 2                                                         00536400
 425  CONTINUE                                                           00536500
      ILLIN = 0                                                          00536600
      RWORK(11) = HU                                                     00536700
      RWORK(12) = H                                                      00536800
      RWORK(13) = TN                                                     00536900
      RWORK(15) = TSW                                                    00537000
      IWORK(11) = NST                                                    00537100
      IWORK(12) = NFE                                                    00537200
      IWORK(13) = NJE                                                    00537300
      IWORK(14) = NQU                                                    00537400
      IWORK(15) = NQ                                                     00537500
      IWORK(19) = MUSED                                                  00537600
      IWORK(20) = METH                                                   00537700
      IWORK(10) = NGE                                                    00537800
      TLAST = T                                                          00537900
      RETURN                                                             00538000
C                                                                        00538100
 430  NTREP = NTREP + 1                                                  00538200
      IF (NTREP .LT. 5) RETURN                                           00538300
      CALL XERRWV(                                                       00538400
     1   8, 301, 1, 0, 0, 0, 1, T, 0.0D0)                                00538500
      GO TO 800                                                          00538600
```

```
C-----------------------------------------------------------------------00538700
C BLOCK H.                                                              00538800
C THE FOLLOWING BLOCK HANDLES ALL UNSUCCESSFUL RETURNS OTHER THAN       00538900
C THOSE FOR ILLEGAL INPUT.  FIRST THE ERROR MESSAGE ROUTINE IS CALLED.  00539000
C IF THERE WAS AN ERROR TEST OR CONVERGENCE TEST FAILURE, IMXER IS SET. 00539100
C THEN Y IS LOADED FROM YH, T IS SET TO TN, AND THE ILLEGAL INPUT       00539200
C COUNTER ILLIN IS SET TO 0.  THE OPTIONAL OUTPUTS ARE LOADED INTO      00539300
C THE WORK ARRAYS BEFORE RETURNING.                                     00539400
C-----------------------------------------------------------------------00539500
C THE MAXIMUM NUMBER OF STEPS WAS TAKEN BEFORE REACHING TOUT. ----------00539600
 500  CALL XERRWV(                                                      00539700
     1   9, 201, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                          00539800
      CALL XERRWV(                                                      00539900
     1   9, 201, 1, 1, MXSTEP, 0, 1, TN, 0.0D0)                        00540000
      ISTATE = -1                                                       00540100
      GO TO 580                                                         00540200
C EWT(I) .LE. 0.0 FOR SOME I (NOT AT START OF PROBLEM). ----------------00540300
 510  EWTI = RWORK(LEWT+I-1)                                            00540400
      CALL XERRWV(                                                      00540500
     1   10, 202, 1, 1, I, 0, 2, TN, EWTI)                             00540600
      ISTATE = -6                                                       00540700
      GO TO 580                                                         00540800
C TOO MUCH ACCURACY REQUESTED FOR MACHINE PRECISION. -------------------00540900
 520  CALL XERRWV(                                                      00541000
     1   11, 203, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                         00541100
      CALL XERRWV(                                                      00541200
     1   11, 203, 1, 0, 0, 0, 2, TN, TOLSF)                            00541300
      RWORK(14) = TOLSF                                                 00541400
      ISTATE = -2                                                       00541500
      GO TO 580                                                         00541600
C KFLAG = -1.  ERROR TEST FAILED REPEATEDLY OR WITH ABS(H) = HMIN. -----00541700
 530  CALL XERRWV(                                                      00541800
     1   12, 204, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                         00541900
      CALL XERRWV(                                                      00542000
     1   12, 204, 1, 0, 0, 0, 2, TN, H)                                00542100
      ISTATE = -4                                                       00542200
      GO TO 560                                                         00542300
C KFLAG = -2.  CONVERGENCE FAILED REPEATEDLY OR WITH ABS(H) = HMIN. ----00542400
 540  CALL XERRWV(                                                      00542500
     1   13, 205, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                         00542600
C     CALL XERRWV(50H      CORRECTOR CONVERGENCE FAILED REPEATEDLY     ,00542700
C    1   50, 205, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                         00542800
      CALL XERRWV(                                                      00542900
     1   13, 205, 1, 0, 0, 0, 2, TN, H)                                00543000
      ISTATE = -5                                                       00543100
      GO TO 560                                                         00543200
C RWORK LENGTH TOO SMALL TO PROCEED. -----------------------------------00543300
 550  CALL XERRWV(                                                      00543400
     1   14, 206, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                         00543500
      CALL XERRWV(                                                      00543600
     1   14, 206, 1, 0, 0, 0, 1, TN, 0.0D0)                            00543700
      ISTATE = -7                                                       00543800
      GO TO 580                                                         00543900
C IWORK LENGTH TOO SMALL TO PROCEED. -----------------------------------00544000
 555  CALL XERRWV(                                                      00544100
     1   15, 207, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                         00544200
      CALL XERRWV(                                                      00544300
     1   15, 207, 1, 0, 0, 0, 1, TN, 0.0D0)                            00544400
      ISTATE = -7                                                       00544500
      GO TO 580                                                         00544600
C COMPUTE IMXER IF RELEVANT. -------------------------------------------00544700
 560  BIG = 0.0D0                                                       00544800
      IMXER = 1                                                         00544900
      DO 570 I = 1,N                                                    00545000
        SIZE = DABS(RWORK(I+LACOR-1)*RWORK(I+LEWT-1))                   00545100
        IF (BIG .GE. SIZE) GO TO 570                                    00545200
        BIG = SIZE                                                      00545300
        IMXER = I                                                       00545400
 570    CONTINUE                                                        00545500
      IWORK(16) = IMXER                                                 00545600
C SET Y VECTOR, T, ILLIN, AND OPTIONAL OUTPUTS. ------------------------00545700
 580  DO 590 I = 1,N                                                    00545800
```

```
 590    Y(I) = RWORK(I+LYH-1)                                          00545900
      T = TN                                                           00546000
      ILLIN = 0                                                        00546100
      RWORK(11) = HU                                                   00546200
      RWORK(12) = H                                                    00546300
      RWORK(13) = TN                                                   00546400
      RWORK(15) = TSW                                                  00546500
      IWORK(11) = NST                                                  00546600
      IWORK(12) = NFE                                                  00546700
      IWORK(13) = NJE                                                  00546800
      IWORK(14) = NQU                                                  00546900
      IWORK(15) = NQ                                                   00547000
      IWORK(19) = MUSED                                                00547100
      IWORK(20) = METH                                                 00547200
      IWORK(10) = NGE                                                  00547300
      TLAST = T                                                        00547400
      RETURN                                                           00547500
C-----------------------------------------------------------------------00547600
C BLOCK I.                                                             00547700
C THE FOLLOWING BLOCK HANDLES ALL ERROR RETURNS DUE TO ILLEGAL INPUT   00547800
C (ISTATE = -3), AS DETECTED BEFORE CALLING THE CORE INTEGRATOR.       00547900
C FIRST THE ERROR MESSAGE ROUTINE IS CALLED.  THEN IF THERE HAVE BEEN  00548000
C 5 CONSECUTIVE SUCH RETURNS JUST BEFORE THIS CALL TO THE SOLVER,      00548100
C THE RUN IS HALTED.                                                   00548200
C-----------------------------------------------------------------------00548300
 601  CALL XERRWV(                                                     00548400
     1   16, 1, 1, 1, ISTATE, 0, 0, 0.0D0, 0.0D0)                      00548500
      GO TO 700                                                        00548600
 602  CALL XERRWV(                                                     00548700
     1   17, 2, 1, 1, ITASK, 0, 0, 0.0D0, 0.0D0)                       00548800
      GO TO 700                                                        00548900
 603  CALL XERRWV(                                                     00549000
     1   18, 3, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                           00549100
      GO TO 700                                                        00549200
 604  CALL XERRWV(                                                     00549300
     1   19, 4, 1, 1, NEQ, 0, 0, 0.0D0, 0.0D0)                         00549400
      GO TO 700                                                        00549500
 605  CALL XERRWV(                                                     00549600
     1   20, 5, 1, 2, N, NEQ, 0, 0.0D0, 0.0D0)                         00549700
      GO TO 700                                                        00549800
 606  CALL XERRWV(                                                     00549900
     1   21, 6, 1, 1, ITOL, 0, 0, 0.0D0, 0.0D0)                        00550000
      GO TO 700                                                        00550100
 607  CALL XERRWV(                                                     00550200
     1   22, 7, 1, 1, IOPT, 0, 0, 0.0D0, 0.0D0)                        00550300
      GO TO 700                                                        00550400
 608  CALL XERRWV(                                                     00550500
     1   23, 8, 1, 1, JT, 0, 0, 0.0D0, 0.0D0)                          00550600
      GO TO 700                                                        00550700
 609  CALL XERRWV(                                                     00550800
     1   24, 9, 1, 2, ML, NEQ, 0, 0.0D0, 0.0D0)                        00550900
      GO TO 700                                                        00551000
 610  CALL XERRWV(                                                     00551100
     1   25, 10, 1, 2, MU, NEQ, 0, 0.0D0, 0.0D0)                       00551200
      GO TO 700                                                        00551300
 611  CALL XERRWV(                                                     00551400
     1   26, 11, 1, 1, IXPR, 0, 0, 0.0D0, 0.0D0)                       00551500
      GO TO 700                                                        00551600
 612  CALL XERRWV(                                                     00551700
     1   27, 12, 1, 1, MXSTEP, 0, 0, 0.0D0, 0.0D0)                     00551800
      GO TO 700                                                        00551900
 613  CALL XERRWV(                                                     00552000
     1   28, 13, 1, 1, MXHNIL, 0, 0, 0.0D0, 0.0D0)                     00552100
      GO TO 700                                                        00552200
 614  CALL XERRWV(                                                     00552300
     1   29, 14, 1, 0, 0, 0, 2, TOUT, T)                               00552400
      CALL XERRWV(                                                     00552500
     1   29, 14, 1, 0, 0, 0, 1, H0, 0.0D0)                             00552600
      GO TO 700                                                        00552700
 615  CALL XERRWV(                                                     00552800
     1   30, 15, 1, 0, 0, 0, 1, HMAX, 0.0D0)                           00552900
      GO TO 700                                                        00553000
```

```
 616  CALL XERRWV(                                                      00553100
     1   31, 16, 1, 0, 0, 0, 1, HMIN, 0.0D0)                           00553200
      GO TO 700                                                        00553300
 617  CALL XERRWV(                                                      00553400
     1   32, 17, 1, 2, LENRW, LRW, 0, 0.0D0, 0.0D0)                    00553500
      GO TO 700                                                        00553600
 618  CALL XERRWV(                                                      00553700
     1   33, 18, 1, 2, LENIW, LIW, 0, 0.0D0, 0.0D0)                    00553800
      GO TO 700                                                        00553900
 619  CALL XERRWV(                                                      00554000
     1   34, 19, 1, 1, I, 0, 1, RTOLI, 0.0D0)                          00554100
      GO TO 700                                                        00554200
 620  CALL XERRWV(                                                      00554300
     1   35, 20, 1, 1, I, 0, 1, ATOLI, 0.0D0)                          00554400
      GO TO 700                                                        00554500
 621  EWTI = RWORK(LEWT+I-1)                                            00554600
      CALL XERRWV(                                                     00554700
     1   36, 21, 1, 1, I, 0, 1, EWTI, 0.0D0)                           00554800
      GO TO 700                                                        00554900
 622  CALL XERRWV(                                                      00555000
     1   37, 22, 1, 0, 0, 0, 2, TOUT, T)                               00555100
      GO TO 700                                                        00555200
 623  CALL XERRWV(                                                      00555300
     1   38, 23, 1, 1, ITASK, 0, 2, TOUT, TP)                          00555400
      GO TO 700                                                        00555500
 624  CALL XERRWV(                                                      00555600
     1   39, 24, 1, 0, 0, 0, 2, TCRIT, TN)                             00555700
      GO TO 700                                                        00555800
 625  CALL XERRWV(                                                      00555900
     1   40, 25, 1, 0, 0, 0, 2, TCRIT, TOUT)                           00556000
      GO TO 700                                                        00556100
 626  CALL XERRWV(                                                      00556200
     1   41, 26, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                          00556300
      CALL XERRWV(                                                     00556400
     1   41, 26, 1, 0, 0, 0, 1, TOLSF, 0.0D0)                          00556500
      RWORK(14) = TOLSF                                                00556600
      GO TO 700                                                        00556700
 627  CALL XERRWV(                                                      00556800
     1   42, 27, 1, 1, ITASK, 0, 1, TOUT, 0.0D0)                       00556900
      GO TO 700                                                        00557000
 628  CALL XERRWV(                                                      00557100
     1   43, 28, 1, 1, MXORDN, 0, 0, 0.0D0, 0.0D0)                     00557200
      GO TO 700                                                        00557300
 629  CALL XERRWV(                                                      00557400
     1   44, 29, 1, 1, MXORDS, 0, 0, 0.0D0, 0.0D0)                     00557500
      GO TO 700                                                        00557600
 630  CALL XERRWV(                                                      00557700
     1   45, 30, 1, 1, NG, 0, 0, 0.0D0, 0.0D0)                         00557800
      GO TO 700                                                        00557900
 631  CALL XERRWV(                                                      00558000
     1   46, 31, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                          00558100
      CALL XERRWV(                                                     00558200
     1   46, 31, 1, 2, NGC, NG, 0, 0.0D0, 0.0D0)                       00558300
      GO TO 700                                                        00558400
 632  CALL XERRWV(                                                      00558500
     1   47, 32, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                          00558600
C     CALL XERRWV(                                                     00558700
C    1   47, 32, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                          00558800
C                                                                      00558900
 700  IF (ILLIN .EQ. 5) GO TO 710                                       00559000
      ILLIN = ILLIN + 1                                                00559100
      TLAST = T                                                        00559200
      ISTATE = -3                                                      00559300
      RETURN                                                           00559400
 710  CALL XERRWV(                                                      00559500
     1   48, 302, 1, 0, 0, 0, 0, 0.0D0, 0.0D0)                         00559600
C                                                                      00559700
 800  CALL XERRWV(                                                      00559800
     1   48, 303, 2, 0, 0, 0, 0, 0.0D0, 0.0D0)                         00559900
      RETURN                                                           00560000
C94341                                                                 00560100
C94341                                                                 00560200
```

```
C----------------------- END SUBROUTINE LSODAR ----------------------   00560300
      END                                                               00560400
      SUBROUTINE STODA (NEQ, Y, YH, NYH, YH1, EWT, SAVF, ACOR,          00560600
     1   WM, IWM, F, JAC, PJAC, SLVS, INDX)                             00560700
      EXTERNAL F, PJAC, SLVS, JAC                                       00560800
      INTEGER NEQ, NYH, IWM                                             00560900
      INTEGER IOWND, IALTH, IPUP, LMAX, MEO, NQNYH, NSLP,               00561000
     1   ICF, IERPJ, IERSL, JCUR, JSTART, KFLAG, L, METH, MITER,        00561100
     2   MAXORD, MAXCOR, MSBP, MXNCF, N, NQ, NST, NFE, NJE, NQU         00561200
      INTEGER IOWND2, ICOUNT, IRFLAG, JTYP, MUSED, MXORDN, MXORDS       00561300
      INTEGER I, I1, IREDO, IRET, J, JB, M, NCF, NEWQ                   00561400
      INTEGER LM1, LM1P1, LM2, LM2P1, NQM1, NQM2                        00561500
      DOUBLE PRECISION Y, YH, YH1, EWT, SAVF, ACOR, WM                  00561600
      DOUBLE PRECISION ROWND,FLONRT,                                    00561700
     1   CONIT, CRATE, EL, ELCO, HOLD, RMAX, TESCO,                     00561800
     2   CCMAX, ELO, H, HMIN, HMXI, HU, RC, TN, UROUND                  00561900
      DOUBLE PRECISION ROWND2, PDEST, PDLAST, RATIO, CM1, CM2,          00562000
     1   PDNORM                                                         00562100
      DOUBLE PRECISION DCON, DDN, DEL, DELP, DSM, DUP, EXDN, EXSM, EXUP,00562200
     1   R, RH, RHDN, RHSM, RHUP, TOLD, VMNORM                          00562300
      DOUBLE PRECISION ALPHA, DM1, DM2, EXM1, EXM2, PDH, PNORM, RATE,   00562400
     1   RH1, RH1IT, RH2, RM, SM1                                       00562500
      CHARACTER RXN                                                     00562600
      INTEGER FLGERR,UNTFLG                                             00562700
      DIMENSION  Y(1), YH(NYH,1), YH1(1), EWT(1), SAVF(1),              00562800
CN                                                                      00562900
     1   ACOR(1), WM(1), IWM(1)                                         00563000
      DIMENSION SM1(12)                                                 00563100
      COMMON /STREAM/FLONRT(6),STRMID(9,10,7),FAO,P,VFLO                00563200
     *       ,IPHASE,MAXCMP,INERT(5)                                    00563300
      COMMON /FLAGS/FLGERR,NFLAG1,UNTFLG(5,3),NFLAG(5),NFLGCV(5)        00563400
     *            , ITRS,ITRT,MAXITR,CTOL                               00563500
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                   00563600
     *       ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS    00563700
     *       ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)        00563800
     *       ,RXN(5,14)                                                 00563900
      COMMON /LSOOO1/ ROWND, CONIT, CRATE, EL(13), ELCO(13,12),   ,     00564000
     1   HOLD, RMAX, TESCO(3,12),                                       00564100
     2   CCMAX, ELO, H, HMIN, HMXI, HU, RC, TN, UROUND, IOWND(14),      00564200
     3   IALTH, IPUP, LMAX, MEO, NQNYH, NSLP,                           00564300
     4   ICF, IERPJ, IERSL, JCUR, JSTART, KFLAG, L, METH, MITER,        00564400
     5   MAXORD, MAXCOR, MSBP, MXNCF, N, NQ, NST, NFE, NJE, NQU         00564500
      COMMON /LSAOO1/ ROWND2, PDEST, PDLAST, RATIO, CM1(12), CM2(5),    00564600
     1   PDNORM,                                                        00564700
     2   IOWND2(3), ICOUNT, IRFLAG, JTYP, MUSED, MXORDN, MXORDS         00564800
      DATA SM1/0.5D0, 0.575D0, 0.55D0, 0.45D0, 0.35D0, 0.25D0,          00564900
     1   0.20D0, 0.15D0, 0.10D0, 0.075D0, 0.050D0, 0.025D0/             00565000
      KFLAG = 0                                                         00565100
      TOLD = TN                                                         00565200
      NCF = 0                                                           00565300
      IERPJ = 0                                                         00565400
      IERSL = 0                                                         00565500
      JCUR = 0                                                          00565600
      ICF = 0                                                           00565700
      IF (JSTART .GT. 0) GO TO 200                                      00565800
      IF (JSTART .EQ. -1) GO TO 100                                     00565900
      IF (JSTART .EQ. -2) GO TO 160                                     00566000
      LMAX = MAXORD + 1                                                 00566100
      NQ = 1                                                            00566200
      L = 2                                                             00566300
      IALTH = 2                                                         00566400
      RMAX = 10000.0D0                                                  00566500
      RC = 0.0D0                                                        00566600
      ELO = 1.0D0                                                       00566700
      CRATE = 0.7D0                                                     00566800
      DELP = 0.0D0                                                      00566900
      HOLD = H                                                          00567000
      NSLP = 0                                                          00567100
      IPUP = MITER                                                      00567200
      IRET = 3                                                          00567300
      ICOUNT = 20                                                       00567400
      IRFLAG = 0                                                        00567500
```

```
      PDEST = 0.0D0                                                     00567600
      PDLAST = 0.0D0                                                    00567700
      RATIO = 5.0D0                                                     00567800
      CALL CFODE (2, ELCO, TESCO)                                       00567900
      DO 10 I = 1,5                                                     00568000
 10     CM2(I) = TESCO(2,I)*ELCO(I+1,I)                                 00568100
      CALL CFODE (1, ELCO, TESCO)                                       00568200
      DO 20 I = 1,12                                                    00568300
 20     CM1(I) = TESCO(2,I)*ELCO(I+1,I)                                 00568400
      GO TO 150                                                         00568500
 100  IPUP = MITER                                                      00568600
      LMAX = MAXORD + 1                                                 00568700
      IF (IALTH .EQ. 1) IALTH = 2                                       00568800
      IF (METH .EQ. MUSED) GO TO 160                                    00568900
      CALL CFODE (METH, ELCO, TESCO)                                    00569000
      IALTH = L                                                         00569100
      IRET = 1                                                          00569200
 150  DO 155 I = 1,L                                                    00569300
 155    EL(I) = ELCO(I,NQ)                                              00569400
      NQNYH = NQ*NYH                                                    00569500
      RC = RC*EL(1)/ELO                                                 00569600
      ELO = EL(1)                                                       00569700
      CONIT = 0.5D0/DFLOAT(NQ+2)                                        00569800
      GO TO (160, 170, 200), IRET                                       00569900
 160  IF (H .EQ. HOLD) GO TO 200                                        00570000
      RH = H/HOLD                                                       00570100
      H = HOLD                                                          00570200
      IREDO = 3                                                         00570300
      GO TO 175                                                         00570400
 170  RH = DMAX1(RH,HMIN/DABS(H))                                       00570500
 175  RH = DMIN1(RH,RMAX)                                               00570600
      RH = RH/DMAX1(1.0D0,DABS(H)*HMXI*RH)                              00570700
      IF (METH .EQ. 2) GO TO 178                                        00570800
      IRFLAG = 0                                                        00570900
      PDH = DMAX1(DABS(H)*PDLAST,0.000001D0)                            00571000
      IF (RH*PDH*1.00001D0 .LT. SM1(NQ)) GO TO 178                      00571100
      RH = SM1(NQ)/PDH                                                  00571200
      IRFLAG = 1                                                        00571300
 178  CONTINUE                                                          00571400
      R = 1.0D0                                                         00571500
      DO 180 J = 2,L                                                    00571600
        R = R*RH                                                        00571700
        DO 180 I = 1,N                                                  00571800
 180      YH(I,J) = YH(I,J)*R                                           00571900
      H = H*RH                                                          00572000
      RC = RC*RH                                                        00572100
C------  STODA                                                          00572200
      IALTH = L                                                         00572300
      IF (IREDO .EQ. 0) GO TO 690                                       00572400
 200  IF (DABS(RC-1.0D0) .GT. CCMAX) IPUP = MITER                       00572500
      IF (NST .GE. NSLP+MSBP) IPUP = MITER                              00572600
      TN = TN + H                                                       00572700
      I1 = NQNYH + 1                                                    00572800
      DO 215 JB = 1,NQ                                                  00572900
        I1 = I1 - NYH                                                   00573000
        DO 210 I = I1,NQNYH                                             00573100
 210      YH1(I) = YH1(I) + YH1(I+NYH)                                  00573200
 215    CONTINUE                                                        00573300
      PNORM = VMNORM (N, YH1, EWT)                                      00573400
 220  M = 0                                                             00573500
      RATE = 0.0D0                                                      00573600
      DEL = 0.0D0                                                       00573700
      DO 230 I = 1,N                                                    00573800
 230    Y(I) = YH(I,1)                                                  00573900
      LINUM1=40331                                                      00574100
      CALL FEX (NEQ, TN, Y, SAVF,INDX)                                  00574200
C-------- STODA                                                         00574300
      NFE = NFE + 1                                                     00574400
      IF (IPUP .LE. 0) GO TO 250                                        00574500
      IPUP = 0                                                          00574600
      RC = 1.0D0                                                        00574700
      NSLP = NST                                                        00574800
```

```
      CRATE = 0.7D0                                                     00574900
      CALL PJAC (NEQ, Y, YH, NYH, EWT, ACOR, SAVF, WM, IWM, F, JAC      00575000
     *           ,INDX)                                                 00575100
      IF (IERPJ .NE. 0) GO TO 430                                       00575200
  250 DO 260 I = 1,N                                                    00575300
  260   ACOR(I) = 0.0D0                                                 00575400
  270 IF (MITER .NE. 0) GO TO 350                                       00575500
      DO 290 I = 1,N                                                    00575600
        SAVF(I) = H*SAVF(I) - YH(I,2)                                   00575700
  290   Y(I) = SAVF(I) - ACOR(I)                                        00575800
      DEL = VMNORM (N, Y, EWT)                                          00575900
      DO 300 I = 1,N                                                    00576000
        Y(I) = YH(I,1) + EL(1)*SAVF(I)                                  00576100
  300   ACOR(I) = SAVF(I)                                               00576200
      GO TO 400                                                         00576300
  350 DO 360 I = 1,N                                                    00576400
  360   Y(I) = H*SAVF(I) - (YH(I,2) + ACOR(I))                          00576500
      CALL SLVS (WM, IWM, Y, SAVF)                                      00576600
      IF (IERSL .LT. 0) GO TO 430                                       00576700
      IF (IERSL .GT. 0) GO TO 410                                       00576800
      DEL = VMNORM (N, Y, EWT)                                          00576900
      DO 380 I = 1,N                                                    00577000
        ACOR(I) = ACOR(I) + Y(I)                                        00577100
  380   Y(I) = YH(I,1) + EL(1)*ACOR(I)                                  00577200
  400 CONTINUE                                                          00577300
      IF (DEL .LE. 100.0D0*PNORM*UROUND) GO TO 450                      00577400
      IF (M .EQ. 0 .AND. METH .EQ. 1) GO TO 405                         00577500
      IF (M .EQ. 0) GO TO 402                                           00577600
      RM = 1024.0D0                                                     00577700
      IF (DEL .LE. 1024.0D0*DELP) RM = DEL/DELP                         00577800
      RATE = DMAX1(RATE,RM)                                             00577900
      CRATE = DMAX1(0.2D0*CRATE,RM)                                     00578000
  402 DCON = DEL*DMIN1(1.0D0,1.5D0*CRATE)/(TESCO(2,NQ)*CONIT)           00578100
      IF (DCON .GT. 1.0D0) GO TO 405                                    00578200
      PDEST = DMAX1(PDEST,RATE/DABS(H*EL(1)))                           00578300
      IF (PDEST .NE. 0.0D0) PDLAST = PDEST                              00578400
      GO TO 450                                                         00578500
  405 CONTINUE                                                          00578600
      M = M + 1                                                         00578700
      IF (M .EQ. MAXCOR) GO TO 410                                      00578800
      IF (M .GE. 2 .AND. DEL .GT. 2.0D0*DELP) GO TO 410                 00578900
      DELP = DEL                                                        00579000
      LINUM1=40811                                                      00579200
      CALL FEX (NEQ, TN, Y, SAVF,INDX)                                  00579300
C-------- STODA                                                        00579400
      NFE = NFE + 1                                                     00579500
      GO TO 270                                                         00579600
  410 IF (MITER .EQ. 0 .OR. JCUR .EQ. 1) GO TO 430                      00579700
      ICF = 1                                                           00579800
      IPUP = MITER                                                      00579900
      GO TO 220                                                         00580000
  430 ICF = 2                                                           00580100
      NCF = NCF + 1                                                     00580200
      RMAX = 2.0D0                                                      00580300
      TN = TOLD                                                         00580400
      I1 = NQNYH + 1                                                    00580500
      DO 445 JB = 1,NQ                                                  00580600
        I1 = I1 - NYH                                                   00580700
        DO 440 I = I1,NQNYH                                             00580800
  440     YH1(I) = YH1(I) - YH1(I+NYH)                                  00580900
  445   CONTINUE                                                        00581000
      IF (IERPJ .LT. 0 .OR. IERSL .LT. 0) GO TO 680                     00581100
      IF (DABS(H) .LE. HMIN*1.00001D0) GO TO 670                        00581200
      IF (NCF .EQ. MXNCF) GO TO 670                                     00581300
      RH = 0.25D0                                                       00581400
      IPUP = MITER                                                      00581500
      IREDO = 1                                                         00581600
      GO TO 170                                                         00581700
  450 JCUR = 0                                                          00581800
      IF (M .EQ. 0) DSM = DEL/TESCO(2,NQ)                               00581900
      IF (M .GT. 0) DSM = VMNORM (N, ACOR, EWT)/TESCO(2,NQ)             00582000
      IF (DSM .GT. 1.0D0) GO TO 500                                     00582100
```

```
      KFLAG = 0                                                          00582200
      IREDO = 0                                                          00582300
      NST = NST + 1                                                      00582400
      HU = H                                                             00582500
      NQU = NQ                                                           00582600
      MUSED = METH                                                       00582700
      DO 460 J = 1,L                                                     00582800
        DO 460 I = 1,N                                                   00582900
  460     YH(I,J) = YH(I,J) + EL(J)*ACOR(I)                              00583000
      ICOUNT = ICOUNT - 1                                                00583100
      IF (ICOUNT .GE. 0) GO TO 488                                       00583200
      IF (METH .EQ. 2) GO TO 480                                         00583300
      IF (NQ .GT. 5) GO TO 488                                           00583400
      IF (DSM .GT. 100.0D0*PNORM*UROUND .AND. PDEST .NE. 0.0D0)          00583500
     1   GO TO 470                                                       00583600
      IF (IRFLAG .EQ. 0) GO TO 488                                       00583700
      RH2 = 2.0D0                                                        00583800
      NQM2 = MIN0(NQ,MXORDS)                                             00583900
      GO TO 478                                                          00584000
  470 CONTINUE                                                           00584100
      EXSM = 1.0D0/DFLOAT(L)                                             00584200
      RH1 = 1.0D0/(1.2D0*DSM**EXSM + 0.0000012D0)                        00584300
      RH1IT = 2.0D0*RH1                                                  00584400
      PDH = PDLAST*DABS(H)                                               00584500
      IF (PDH*RH1 .GT. 0.00001D0) RH1IT = SM1(NQ)/PDH                    00584600
      RH1 = DMIN1(RH1,RH1IT)                                             00584700
      IF (NQ .LE. MXORDS) GO TO 474                                      00584800
         NQM2 = MXORDS                                                   00584900
         LM2 = MXORDS + 1                                                00585000
         EXM2 = 1.0D0/DFLOAT(LM2)                                        00585100
         LM2P1 = LM2 + 1                                                 00585200
         DM2 = VMNORM (N, YH(1,LM2P1), EWT)/CM2(MXORDS)                  00585300
         RH2 = 1.0D0/(1.2D0*DM2**EXM2 + 0.0000012D0)                     00585400
         GO TO 476                                                       00585500
  474 DM2 = DSM*(CM1(NQ)/CM2(NQ))                                        00585600
      RH2 = 1.0D0/(1.2D0*DM2**EXSM + 0.0000012D0)                        00585700
      NQM2 = NQ                                                          00585800
  476 CONTINUE                                                           00585900
      IF (RH2 .LT. RATIO*RH1) GO TO 488                                  00586000
  478 RH = RH2                                                           00586100
      ICOUNT = 20                                                        00586200
      METH = 2                                                           00586300
      MITER = JTYP                                                       00586400
      PDLAST = 0.0D0                                                     00586500
      NQ = NQM2                                                          00586600
      L = NQ + 1                                                         00586700
      GO TO 170                                                          00586800
  480 CONTINUE                                                           00586900
      EXSM = 1.0D0/DFLOAT(L)                                             00587000
      IF (MXORDN .GE. NQ) GO TO 484                                      00587100
         NQM1 = MXORDN                                                   00587200
         LM1 = MXORDN + 1                                                00587300
         EXM1 = 1.0D0/DFLOAT(LM1)                                        00587400
         LM1P1 = LM1 + 1                                                 00587500
         DM1 = VMNORM (N, YH(1,LM1P1), EWT)/CM1(MXORDN)                  00587600
         RH1 = 1.0D0/(1.2D0*DM1**EXM1 + 0.0000012D0)                     00587700
         GO TO 486                                                       00587800
  484 DM1 = DSM*(CM2(NQ)/CM1(NQ))                                        00587900
      RH1 = 1.0D0/(1.2D0*DM1**EXSM + 0.0000012D0)                        00588000
      NQM1 = NQ                                                          00588100
      EXM1 = EXSM                                                        00588200
  486 RH1IT = 2.0D0*RH1                                                  00588300
      PDH = PDNORM*DABS(H)                                               00588400
      IF (PDH*RH1 .GT. 0.00001D0) RH1IT = SM1(NQM1)/PDH                  00588500
      RH1 = DMIN1(RH1,RH1IT)                                             00588600
      RH2 = 1.0D0/(1.2D0*DSM**EXSM + 0.0000012D0)                        00588700
      IF (RH1*RATIO .LT. 5.0D0*RH2) GO TO 488                            00588800
      ALPHA = DMAX1(0.001D0,RH1)                                         00588900
      DM1 = (ALPHA**EXM1)*DM1                                            00589000
      IF (DM1 .LE. 1000.0D0*UROUND*PNORM) GO TO 488                      00589100
      RH = RH1                                                           00589200
      ICOUNT = 20                                                        00589300
```

```
      METH = 1                                                          00589400
      MITER = 0                                                         00589500
      PDLAST = 0.0D0                                                    00589600
      NQ = NQM1                                                         00589700
      L = NQ + 1                                                        00589800
      GO TO 170                                                         00589900
  488 CONTINUE                                                          00590000
      IALTH = IALTH - 1                                                 00590100
      IF (IALTH .EQ. 0) GO TO 520                                       00590200
      IF (IALTH .GT. 1) GO TO 700                                       00590300
      IF (L .EQ. LMAX) GO TO 700                                        00590400
      DO 490 I = 1,N                                                    00590500
  490   YH(I,LMAX) = ACOR(I)                                            00590600
      GO TO 700                                                         00590700
  500 KFLAG = KFLAG - 1                                                 00590800
      TN = TOLD                                                         00590900
      I1 = NQNYH + 1                                                    00591000
      DO 515 JB = 1,NQ                                                  00591100
        I1 = I1 - NYH                                                   00591200
        DO 510 I = I1,NQNYH                                             00591300
  510     YH1(I) = YH1(I) - YH1(I+NYH)                                  00591400
  515   CONTINUE                                                        00591500
      RMAX = 2.0D0                                                      00591600
      IF (DABS(H) .LE. HMIN*1.00001D0) GO TO 660                        00591700
      IF (KFLAG .LE. -3) GO TO 640                                      00591800
      IREDO = 2                                                         00591900
      RHUP = 0.0D0                                                      00592000
      GO TO 540                                                         00592100
  520 RHUP = 0.0D0                                                      00592200
      IF (L .EQ. LMAX) GO TO 540                                        00592300
      DO 530 I = 1,N                                                    00592400
  530   SAVF(I) = ACOR(I) - YH(I,LMAX)                                  00592500
      DUP = VMNORM (N, SAVF, EWT)/TESCO(3,NQ)                           00592600
      EXUP = 1.0D0/DFLOAT(L+1)                                          00592700
      RHUP = 1.0D0/(1.4D0*DUP**EXUP + 0.0000014D0)                      00592800
  540 EXSM = 1.0D0/DFLOAT(L)                                            00592900
      RHSM = 1.0D0/(1.2D0*DSM**EXSM + 0.0000012D0)                      00593000
      RHDN = 0.0D0                                                      00593100
      IF (NQ .EQ. 1) GO TO 550                                          00593200
      DDN = VMNORM (N, YH(1,L), EWT)/TESCO(1,NQ)                        00593300
      EXDN = 1.0D0/DFLOAT(NQ)                                           00593400
      RHDN = 1.0D0/(1.3D0*DDN**EXDN + 0.0000013D0)                      00593500
  550 IF (METH .EQ. 2) GO TO 560                                        00593600
      PDH = DMAX1(DABS(H)*PDLAST,0.000001D0)                            00593700
      IF (L .LT. LMAX) RHUP = DMIN1(RHUP,SM1(L)/PDH)                    00593800
      RHSM = DMIN1(RHSM,SM1(NQ)/PDH)                                    00593900
      IF (NQ .GT. 1) RHDN = DMIN1(RHDN,SM1(NQ-1)/PDH)                   00594000
      PDEST = 0.0D0                                                     00594100
  560 IF (RHSM .GE. RHUP) GO TO 570                                     00594200
      IF (RHUP .GT. RHDN) GO TO 590                                     00594300
      GO TO 580                                                         00594400
  570 IF (RHSM .LT. RHDN) GO TO 580                                     00594500
      NEWQ = NQ                                                         00594600
      RH = RHSM                                                         00594700
      GO TO 620                                                         00594800
  580 NEWQ = NQ - 1                                                     00594900
      RH = RHDN                                                         00595000
      IF (KFLAG .LT. 0 .AND. RH .GT. 1.0D0) RH = 1.0D0                  00595100
      GO TO 620                                                         00595200
  590 NEWQ = L                                                          00595300
      RH = RHUP                                                         00595400
      IF (RH .LT. 1.1D0) GO TO 610                                      00595500
      R = EL(L)/DFLOAT(L)                                               00595600
      DO 600 I = 1,N                                                    00595700
  600   YH(I,NEWQ+1) = ACOR(I)*R                                        00595800
      GO TO 630                                                         00595900
  610 IALTH = 3                                                         00596000
      GO TO 700                                                         00596100
  620 IF (METH .EQ. 2) GO TO 622                                        00596200
      IF (RH*PDH*1.00001D0 .GE. SM1(NEWQ)) GO TO 625                    00596300
  622 IF (KFLAG .EQ. 0 .AND. RH .LT. 1.1D0) GO TO 610                   00596400
  625 IF (KFLAG .LE. -2) RH = DMIN1(RH,0.2D0)                           00596500
```

```
      IF (NEWQ .EQ. NQ) GO TO 170                                       00596600
 630  NQ = NEWQ                                                         00596700
      L = NQ + 1                                                        00596800
      IRET = 2                                                          00596900
      GO TO 150                                                         00597000
 640  IF (KFLAG .EQ. -10) GO TO 660                                     00597100
      RH = 0.1D0                                                        00597200
      RH = DMAX1(HMIN/DABS(H),RH)                                       00597300
      H = H*RH                                                          00597400
      DO 645 I = 1,N                                                    00597500
 645    Y(I) = YH(I,1)                                                  00597600
      CALL FEX (NEQ, TN, Y, SAVF,INDX)                                  00597900
C------  STODA                                                          00598000
      NFE = NFE + 1                                                     00598100
      DO 650 I = 1,N                                                    00598200
 650    YH(I,2) = H*SAVF(I)                                             00598300
      IPUP = MITER                                                      00598400
      IALTH = 5                                                         00598500
      IF (NQ .EQ. 1) GO TO 200                                          00598600
      NQ = 1                                                            00598700
      L = 2                                                             00598800
      IRET = 3                                                          00598900
      GO TO 150                                                         00599000
 660  KFLAG = -1                                                        00599100
      GO TO 720                                                         00599200
 670  KFLAG = -2                                                        00599300
      GO TO 720                                                         00599400
 680  KFLAG = -3                                                        00599500
      GO TO 720                                                         00599600
 690  RMAX = 10.0D0                                                     00599700
 700  R = 1.0D0/TESCO(2,NQU)                                            00599800
      DO 710 I = 1,N                                                    00599900
 710    ACOR(I) = ACOR(I)*R                                             00600000
 720  HOLD = H                                                          00600100
      JSTART = 1                                                        00600200
      RETURN                                                            00600300
C----------------------- END OF SUBROUTINE STODA -----------------------00600400
      END                                                               00600500
      SUBROUTINE INTDY (T, K, YH, NYH, DKY, IFLAG)                      00600800
CLLL. OPTIMIZE                                                          00600900
      INTEGER K, NYH, IFLAG                                             00601000
      INTEGER IOWND, IOWNS,                                             00601100
     1   ICF, IERPJ, IERSL, JCUR, JSTART, KFLAG, L, METH, MITER,        00601200
     2   MAXORD, MAXCOR, MSBP, MXNCF, N, NQ, NST, NFE, NJE, NQU         00601300
      INTEGER I, IC, J, JB, JB2, JJ, JJ1, JP1                           00601400
      DOUBLE PRECISION T, YH, DKY                                       00601500
      DOUBLE PRECISION ROWND, ROWNS,                                    00601600
     1   CCMAX, ELO, H, HMIN, HMXI, HU, RC, TN, UROUND                  00601700
      DOUBLE PRECISION C, R, S, TP                                      00601800
      DIMENSION YH(NYH,1), DKY(1)                                       00601900
      COMMON /LS0001/ ROWND, ROWNS(209),                                00602000
     2   CCMAX, ELO, H, HMIN, HMXI, HU, RC, TN, UROUND,                 00602100
     3   IOWND(14), IOWNS(6),                                          00602200
     4   ICF, IERPJ, IERSL, JCUR, JSTART, KFLAG, L, METH, MITER,        00602300
     5   MAXORD, MAXCOR, MSBP, MXNCF, N, NQ, NST, NFE, NJE, NQU         00602400
C-----------------------------------------------------------------------00602500
C15551                                                                  00602600
C15731                                                                  00602700
C-----------------------------------------------------------------------00602800
      IFLAG = 0                                                         00602900
      IF (K .LT. 0 .OR. K .GT. NQ) GO TO 80                             00603000
      TP = TN - HU*(1.0D0 + 100.0D0*UROUND)                             00603100
      IF ((T-TP)*(T-TN) .GT. 0.0D0) GO TO 90                            00603200
C                                                                       00603300
      S = (T - TN)/H                                                    00603400
      IC = 1                                                            00603500
      IF (K .EQ. 0) GO TO 15                                            00603600
      JJ1 = L - K                                                       00603700
      DO 10 JJ = JJ1,NQ                                                 00603800
 10     IC = IC*JJ                                                     00603900
 15   C = DFLOAT(IC)                                                    00604000
      DO 20 I = 1,N                                                     00604100
```

```
 20     DKY(I) = C*YH(I,L)                                              00604200
      IF (K .EQ. NQ) GO TO 55                                           00604300
      JB2 = NQ - K                                                      00604400
      DO 50 JB = 1,JB2                                                  00604500
        J = NQ - JB                                                     00604600
        JP1 = J + 1                                                     00604700
        IC = 1                                                          00604800
        IF (K .EQ. 0) GO TO 35                                          00604900
        JJ1 = JP1 - K                                                   00605000
        DO 30 JJ = JJ1,J                                                00605100
 30       IC = IC*JJ                                                    00605200
 35     C = DFLOAT(IC)                                                  00605300
        DO 40 I = 1,N                                                   00605400
 40       DKY(I) = C*YH(I,JP1) + S*DKY(I)                               00605500
 50     CONTINUE                                                        00605600
      IF (K .EQ. 0) RETURN                                              00605700
 55   R = H**(-K)                                                       00605800
      DO 60 I = 1,N                                                     00605900
 60     DKY(I) = R*DKY(I)                                               00606000
      RETURN                                                            00606100
C                                                                       00606200
 80   CALL XERRWV(                                                      00606300
     1   49, 51, 1, 1, K, 0, 0, 0.0D0, 0.0D0)                           00606400
      IFLAG = -1                                                        00606500
      RETURN                                                            00606600
 90   CALL XERRWV(                                                      00606700
     1   50, 52, 1, 0, 0, 0, 1, T, 0.0D0)                               00606800
      CALL XERRWV(                                                      00606900
     1   50, 52, 1, 0, 0, 0, 2, TP, TN)                                 00607000
      IFLAG = -2                                                        00607100
      RETURN                                                            00607200
C----------------------- END SUBROUTINE INTDY -----------------------   00607300
      END                                                               00607400
      SUBROUTINE CFODE (METH, ELCO, TESCO)                              00607600
CLLL. OPTIMIZE                                                          00607700
      INTEGER METH                                                      00607800
      INTEGER I, IB, NQ, NQM1, NQP1                                     00607900
      DOUBLE PRECISION ELCO, TESCO                                      00608000
      DOUBLE PRECISION AGAMQ, FNQ, FNQM1, PC, PINT, RAGQ,               00608100
     1   RQFAC, RQ1FAC, TSIGN, XPIN                                     00608200
      DIMENSION ELCO(13,12), TESCO(3,12)                                00608300
      DIMENSION PC(12)                                                  00608800
      GO TO (100, 200), METH                                            00609100
 100  ELCO(1,1) = 1.0D0                                                00609300
      ELCO(2,1) = 1.0D0                                                 00609400
      TESCO(1,1) = 0.0D0                                                00609500
      TESCO(2,1) = 2.0D0                                                00609600
      TESCO(1,2) = 1.0D0                                                00609700
      TESCO(3,12) = 0.0D0                                               00609800
      PC(1) = 1.0D0                                                     00609900
      RQFAC = 1.0D0                                                     00610000
      DO 140 NQ = 2,12                                                  00610100
C-----------------------------------------------------------------------00610200
C THE PC ARRAY WILL CONTAIN THE COEFFICIENTS OF THE POLYNOMIAL          00610300
C     P(X) = (X+1)*(X+2)*...*(X+NQ-1).                                  00610400
C INITIALLY, P(X) = 1.                                                  00610500
C-----------------------------------------------------------------------00610600
        RQ1FAC = RQFAC                                                  00610700
        RQFAC = RQFAC/DFLOAT(NQ)                                        00610800
        NQM1 = NQ - 1                                                   00610900
        FNQM1 = DFLOAT(NQM1)                                            00611000
        NQP1 = NQ + 1                                                   00611100
C FORM COEFFICIENTS OF P(X)*(X+NQ-1). ---------------------------------00611200
        PC(NQ) = 0.0D0                                                  00611300
        DO 110 IB = 1,NQM1                                              00611400
          I = NQP1 - IB                                                 00611500
 110      PC(I) = PC(I-1) + FNQM1*PC(I)                                 00611600
        PC(1) = FNQM1*PC(1)                                             00611700
C COMPUTE INTEGRAL, -1 TO 0, OF P(X) AND X*P(X). ----------------------00611800
        PINT = PC(1)                                                    00611900
        XPIN = PC(1)/2.0D0                                              00612000
        TSIGN = 1.0D0                                                   00612100
```

```
        DO 120 I = 2,NQ                                                    00612200
          TSIGN = -TSIGN                                                   00612300
          PINT = PINT + TSIGN*PC(I)/DFLOAT(I)                              00612400
 120      XPIN = XPIN + TSIGN*PC(I)/DFLOAT(I+1)                            00612500
C STORE COEFFICIENTS IN ELCO AND TESCO. ------------------------------00612600
        ELCO(1,NQ) = PINT*RQ1FAC                                           00612700
        ELCO(2,NQ) = 1.0D0                                                 00612800
        DO 130 I = 2,NQ                                                    00612900
 130      ELCO(I+1,NQ) = RQ1FAC*PC(I)/DFLOAT(I)                            00613000
        AGAMQ = RQFAC*XPIN                                                 00613100
        RAGQ = 1.0D0/AGAMQ                                                 00613200
        TESCO(2,NQ) = RAGQ                                                 00613300
        IF (NQ .LT. 12) TESCO(1,NQP1) = RAGQ*RQFAC/DFLOAT(NQP1)            00613400
        TESCO(3,NQM1) = RAGQ                                               00613500
 140    CONTINUE                                                           00613600
      RETURN                                                               00613700
C                                                                          00613800
 200  PC(1) = 1.0D0                                                        00613900
      RQ1FAC = 1.0D0                                                       00614000
      DO 230 NQ = 1,5                                                      00614100
C-------------------------------------------------------------------------00614200
C THE PC ARRAY WILL CONTAIN THE COEFFICIENTS OF THE POLYNOMIAL             00614300
C     P(X) = (X+1)*(X+2)*...*(X+NQ).                                       00614400
C INITIALLY, P(X) = 1.                                                     00614500
C-------------------------------------------------------------------------00614600
        FNQ = DFLOAT(NQ)                                                   00614700
        NQP1 = NQ + 1                                                      00614800
C FORM COEFFICIENTS OF P(X)*(X+NQ). ----------------------------------00614900
        PC(NQP1) = 0.0D0                                                   00615000
        DO 210 IB = 1,NQ                                                   00615100
          I = NQ + 2 - IB                                                  00615200
 210      PC(I) = PC(I-1) + FNQ*PC(I)                                      00615300
        PC(1) = FNQ*PC(1)                                                  00615400
C STORE COEFFICIENTS IN ELCO AND TESCO. ------------------------------00615500
        DO 220 I = 1,NQP1                                                  00615600
 220      ELCO(I,NQ) = PC(I)/PC(2)                                         00615700
        ELCO(2,NQ) = 1.0D0                                                 00615800
        TESCO(1,NQ) = RQ1FAC                                               00615900
        TESCO(2,NQ) = DFLOAT(NQP1)/ELCO(1,NQ)                              00616000
        TESCO(3,NQ) = DFLOAT(NQ+2)/ELCO(1,NQ)                              00616100
        RQ1FAC = RQ1FAC/FNQ                                                00616200
 230    CONTINUE                                                           00616300
      RETURN                                                               00616400
C----------------------- END SUBROUTINE CFODE -----------------------    00616500
      END                                                                  00616600
      SUBROUTINE SOLSY (WM, IWM, X, TEM)                                   00616800
CLLL. OPTIMIZE                                                             00616900
      INTEGER IWM                                                          00617000
      INTEGER IOWND, IOWNS,                                                00617100
     1   ICF, IERPJ, IERSL, JCUR, JSTART, KFLAG, L, METH, MITER,           00617200
     2   MAXORD, MAXCOR, MSBP, MXNCF, N, NQ, NST, NFE, NJE, NQU            00617300
      INTEGER I, MEBAND, ML, MU                                            00617400
      DOUBLE PRECISION WM, X, TEM                                          00617500
      DOUBLE PRECISION ROWND, ROWNS,                                       00617600
     1   CCMAX, ELO, H, HMIN, HMXI, HU, RC, TN, UROUND                     00617700
      DOUBLE PRECISION DI, HLO, PHLO, R                                    00617800
      DIMENSION WM(1), IWM(1), X(1), TEM(1)                                00617900
      COMMON /LS0001/ ROWND, ROWNS(209),                                   00618000
     2   CCMAX, ELO, H, HMIN, HMXI, HU, RC, TN, UROUND,                    00618100
     3   IOWND(14), IOWNS(6),                                              00618200
     4   ICF, IERPJ, IERSL, JCUR, JSTART, KFLAG, L, METH, MITER,           00618300
     5   MAXORD, MAXCOR, MSBP, MXNCF, N, NQ, NST, NFE, NJE, NQU            00618400
C-------------------------------------------------------------------------00618500
C23920                                                                     00618600
C24140                                                                     00618700
C-------------------------------------------------------------------------00618800
      IERSL = 0                                                            00619000
      GO TO (100, 100, 300, 400, 400), MITER                               00619100
 100  CALL DGESL (WM(3), N, N, IWM(21), X, 0)                              00619200
      RETURN                                                               00619300
C                                                                          00619400
 300  PHLO = WM(2)                                                         00619500
```

```
      HLO = H*ELO                                                       00619600
      WM(2) = HLO                                                       00619700
      IF (HLO .EQ. PHLO) GO TO 330                                      00619800
      R = HLO/PHLO                                                      00619900
      DO 320 I = 1,N                                                    00620000
        DI = 1.0D0 - R*(1.0D0 - 1.0D0/WM(I+2))                          00620100
        IF (DABS(DI) .EQ. 0.0D0) GO TO 390                              00620200
 320    WM(I+2) = 1.0D0/DI                                              00620300
 330  DO 340 I = 1,N                                                    00620400
 340    X(I) = WM(I+2)*X(I)                                             00620500
      RETURN                                                            00620600
 390  IERSL = 1                                                         00620700
      RETURN                                                            00620800
C                                                                       00620900
 400  ML = IWM(1)                                                       00621000
      MU = IWM(2)                                                       00621100
      MEBAND = 2*ML + MU + 1                                            00621200
      CALL DGBSL (WM(3), MEBAND, N, ML, MU, IWM(21), X, 0)              00621300
      RETURN                                                            00621400
C----------------------- END OF SUBROUTINE SOLSY -----------------------00621500
      END                                                               00621600
      SUBROUTINE EWSET (N, ITOL, RTOL, ATOL, YCUR, EWT)                 00621800
CLLL. OPTIMIZE                                                          00621900
C-----------------------------------------------------------------------00622000
C THIS SUBROUTINE SETS THE ERROR WEIGHT VECTOR EWT ACCORDING TO         00622100
C     EWT(I) = RTOL(I)*ABS(YCUR(I)) + ATOL(I),  I = 1,...,N,            00622200
C WITH THE SUBSCRIPT ON RTOL AND/OR ATOL POSSIBLY REPLACED BY 1 ABOVE,  00622300
C DEPENDING ON THE VALUE OF ITOL.                                       00622400
C-----------------------------------------------------------------------00622500
      INTEGER N, ITOL                                                   00622600
      INTEGER I                                                         00622700
      DOUBLE PRECISION RTOL, ATOL, YCUR, EWT                            00622800
      DOUBLE PRECISION ATOLI, RTOLI                                     00622900
      DIMENSION RTOL(1), ATOL(1), YCUR(N), EWT(N)                       00623000
      RTOLI = RTOL(1)                                                   00623200
      ATOLI = ATOL(1)                                                   00623300
      DO 10 I = 1,N                                                     00623400
        IF (ITOL .GE. 3) RTOLI = RTOL(I)                                00623500
        IF (ITOL .EQ. 2 .OR. ITOL .EQ. 4) ATOLI = ATOL(I)               00623600
        EWT(I) = RTOLI*DABS(YCUR(I)) + ATOLI                            00623700
 10     CONTINUE                                                        00623800
      RETURN                                                            00623900
C----------------------- END OF SUBROUTINE EWSET -----------------------00624000
      END                                                               00624100
      DOUBLE PRECISION FUNCTION VNORM (N, V, W)                         00624200
CLLL. OPTIMIZE                                                          00624300
C-----------------------------------------------------------------------00624400
C THIS FUNCTION ROUTINE COMPUTES THE WEIGHTED ROOT-MEAN-SQUARE NORM     00624500
C OF THE VECTOR OF LENGTH N CONTAINED IN THE ARRAY V, WITH WEIGHTS      00624600
C CONTAINED IN THE ARRAY W OF LENGTH N..                                00624700
C   VNORM = SQRT( (1/N) * SUM( V(I)*W(I) )**2 )                        00624800
C-----------------------------------------------------------------------00624900
      INTEGER N,   I                                                    00625000
      DOUBLE PRECISION V, W,   SUM                                      00625100
      DIMENSION V(N), W(N)                                              00625200
      SUM = 0.0D0                                                       00625300
      DO 10 I = 1,N                                                     00625400
 10     SUM = SUM + (V(I)*W(I))**2                                      00625500
      VNORM = DSQRT(SUM/DFLOAT(N))                                      00625600
      RETURN                                                            00625700
      END                                                               00625800
C----------------------- END OF FUNCTION VNORM -------------------------00625900
      SUBROUTINE JDUM (NEQ, TN, Y, I, II, WM, N)                        00626200
      DOUBLE PRECISION TN, Y                                            00626300
      DIMENSION Y(1), WM(1)                                             00626400
CN                                                                      00626500
      RETURN                                                            00626700
      END                                                               00626800
      SUBROUTINE JAC(NEQ, TN, Y, I, II, WM, N)                          00626900
      DOUBLE PRECISION WM, TN, Y                                        00627000
      DIMENSION Y(1)                                                    00627100
      RETURN                                                            00627400
```

```
      END                                                               00627500
      SUBROUTINE PRJA (NEQ, Y, YH, NYH, EWT, FTEM, SAVF, WM, IWM,       00627600
     1   F, JAC,INDX)                                                   00627700
CLLL. OPTIMIZE                                                          00627800
      EXTERNAL F, JAC                                                   00627900
      INTEGER NEQ, NYH, IWM                                             00628000
      INTEGER IOWND, IOWNS,                                             00628100
     1   ICF, IERPJ, IERSL, JCUR, JSTART, KFLAG, L, METH, MITER,        00628200
     2   MAXORD, MAXCOR, MSBP, MXNCF, N, NQ, NST, NFE, NJE, NQU         00628300
      INTEGER IOWND2, IOWNS2, JTYP, MUSED, MXORDN, MXORDS               00628400
      INTEGER I, I1, I2, IER, II, J, J1, JJ, LENP,                      00628500
     1   MBA, MBAND, MEB1, MEBAND, ML, ML3, MU                          00628600
      DOUBLE PRECISION Y, YH, EWT, FTEM, SAVF, WM                       00628700
      DOUBLE PRECISION ROWND, ROWNS,                                    00628800
     1   CCMAX, ELO, H, HMIN, HMXI, HU, RC, TN, UROUND                  00628900
      DOUBLE PRECISION ROWND2, ROWNS2, PDNORM                           00629000
      DOUBLE PRECISION CON, DI, FAC, HLO, R, RO, SRUR, YI, YJ, YJJ,     00629100
     1   VMNORM, FNORM, BNORM                                           00629200
      DIMENSION Y(1), YH(NYH,1), EWT(1), FTEM(1), SAVF(1),              00629300
CN                                                                      00629400
     1   WM(1), IWM(1)                                                  00629500
      COMMON /LS0001/ ROWND, ROWNS(209),                                00629600
     2   CCMAX, ELO, H, HMIN, HMXI, HU, RC, TN, UROUND,                 00629700
     3   IOWND(14), IOWNS(6),                                          00629800
     4   ICF, IERPJ, IERSL, JCUR, JSTART, KFLAG, L, METH, MITER,        00629900
     5   MAXORD, MAXCOR, MSBP, MXNCF, N, NQ, NST, NFE, NJE, NQU         00630000
      COMMON /LSA001/ ROWND2, ROWNS2(20), PDNORM,                       00630100
     1   IOWND2(3), IOWNS2(2), JTYP, MUSED, MXORDN, MXORDS              00630200
C-----------------------------------------------------------------------00630300
C73020                                                                  00630400
C73341                                                                  00630500
C-----------------------------------------------------------------------00630600
      NJE = NJE + 1                                                     00630800
      IERPJ = 0                                                         00630900
      JCUR = 1                                                          00631000
      HLO = H*ELO                                                       00631100
      GO TO (100, 200, 300, 400, 500), MITER                            00631200
C IF MITER = 1, CALL JAC AND MULTIPLY BY SCALAR. ----------------------00631300
 100  LENP = N*N                                                        00631400
      DO 110 I = 1,LENP                                                 00631500
 110    WM(I+2) = 0.0D0                                                 00631600
      CALL JAC (NEQ, TN, Y, 0, 0, WM(3), N)                             00631700
      CON = -HLO                                                        00631800
      DO 120 I = 1,LENP                                                 00631900
 120    WM(I+2) = WM(I+2)*CON                                           00632000
      GO TO 240                                                         00632100
C IF MITER = 2, MAKE N CALLS TO F TO APPROXIMATE J. -------------------00632200
 200  FAC = VMNORM (N, SAVF, EWT)                                       00632300
      RO = 1000.0D0*DABS(H)*UROUND*DFLOAT(N)*FAC                        00632400
      IF (RO .EQ. 0.0D0) RO = 1.0D0                                     00632500
      SRUR = WM(1)                                                      00632600
      J1 = 2                                                            00632700
      DO 230 J = 1,N                                                    00632800
        YJ = Y(J)                                                       00632900
        R = DMAX1(SRUR*DABS(YJ),RO/EWT(J))                              00633000
        Y(J) = Y(J) + R                                                 00633100
        FAC = -HLO/R                                                    00633200
        CALL FEX (NEQ, TN, Y, FTEM,INDX)                                00633300
C------  PRJA                                                           00633400
        DO 220 I = 1,N                                                  00633600
 220      WM(I+J1) = (FTEM(I) - SAVF(I))*FAC                            00633700
        Y(J) = YJ                                                       00633800
        J1 = J1 + N                                                     00633900
 230    CONTINUE                                                        00634000
      NFE = NFE + N                                                     00634100
 240  CONTINUE                                                          00634200
C COMPUTE NORM OF JACOBIAN. --------------------------------------------00634300
      PDNORM = FNORM (N, WM(3), EWT)/DABS(HLO)                          00634400
C ADD IDENTITY MATRIX. -------------------------------------------------00634500
      J = 3                                                             00634600
      DO 250 I = 1,N                                                    00634700
        WM(J) = WM(J) + 1.0D0                                           00634800
```

```
 250    J = J + (N + 1)                                                  00634900
C DO LU DECOMPOSITION ON P. ----------------------------------------------00635000
      CALL DGEFA (WM(3), N, N, IWM(21), IER)                            00635100
      IF (IER .NE. 0) IERPJ = 1                                         00635200
      RETURN                                                            00635300
C DUMMY BLOCK ONLY, SINCE MITER IS NEVER 3 IN THIS ROUTINE. ------------00635400
 300  RETURN                                                            00635500
C IF MITER = 4, CALL JAC AND MULTIPLY BY SCALAR. ------------------------00635600
 400  ML = IWM(1)                                                       00635700
      MU = IWM(2)                                                       00635800
      ML3 = ML + 3                                                      00635900
      MBAND = ML + MU + 1                                               00636000
      MEBAND = MBAND + ML                                               00636100
      LENP = MEBAND*N                                                   00636200
      DO 410 I = 1,LENP                                                 00636300
 410    WM(I+2) = 0.0D0                                                 00636400
      CALL JAC (NEQ, TN, Y, ML, MU, WM(ML3), MEBAND)                    00636500
      CON = -HLO                                                        00636600
      DO 420 I = 1,LENP                                                 00636700
 420    WM(I+2) = WM(I+2)*CON                                           00636800
      GO TO 570                                                         00636900
C IF MITER = 5, MAKE MBAND CALLS TO F TO APPROXIMATE J. ----------------00637000
 500  ML = IWM(1)                                                       00637100
      MU = IWM(2)                                                       00637200
      MBAND = ML + MU + 1                                               00637300
      MBA = MIN0(MBAND,N)                                               00637400
      MEBAND = MBAND + ML                                               00637500
      MEB1 = MEBAND - 1                                                 00637600
      SRUR = WM(1)                                                      00637700
      FAC = VMNORM (N, SAVF, EWT)                                       00637800
      R0 = 1000.0D0*DABS(H)*UROUND*DFLOAT(N)*FAC                        00637900
      IF (R0 .EQ. 0.0D0) R0 = 1.0D0                                     00638000
      DO 560 J = 1,MBA                                                  00638100
        DO 530 I = J,N,MBAND                                            00638200
          YI = Y(I)                                                     00638300
          R = DMAX1(SRUR*DABS(YI),R0/EWT(I))                            00638400
 530      Y(I) = Y(I) + R                                               00638500
        CALL FEX (NEQ, TN, Y, FTEM,INDX)                                00638600
C------ PRJA                                                           00638700
        DO 550 JJ = J,N,MBAND                                           00638900
          Y(JJ) = YH(JJ,1)                                              00639000
          YJJ = Y(JJ)                                                   00639100
          R = DMAX1(SRUR*DABS(YJJ),R0/EWT(JJ))                          00639200
          FAC = -HLO/R                                                  00639300
          I1 = MAX0(JJ-MU,1)                                            00639400
          I2 = MIN0(JJ+ML,N)                                            00639500
          II = JJ*MEB1 - ML + 2                                         00639600
          DO 540 I = I1,I2                                              00639700
 540        WM(II+I) = (FTEM(I) - SAVF(I))*FAC                          00639800
 550      CONTINUE                                                      00639900
 560    CONTINUE                                                        00640000
      NFE = NFE + MBA                                                   00640100
 570  CONTINUE                                                          00640200
C COMPUTE NORM OF JACOBIAN. ----------------------------------------------00640300
      PDNORM = BNORM (N, WM(3), MEBAND, ML, MU, EWT)/DABS(HLO)          00640400
C ADD IDENTITY MATRIX. ---------------------------------------------------00640500
      II = MBAND + 2                                                    00640600
      DO 580 I = 1,N                                                    00640700
        WM(II) = WM(II) + 1.0D0                                         00640800
 580    II = II + MEBAND                                                00640900
C DO LU DECOMPOSITION OF P. ----------------------------------------------00641000
      CALL DGBFA (WM(3), MEBAND, N, ML, MU, IWM(21), IER)               00641100
      IF (IER .NE. 0) IERPJ = 1                                         00641200
      RETURN                                                            00641300
C----------------------- END OF SUBROUTINE PRJA -------------------------00641400
      END                                                               00641500
      DOUBLE PRECISION FUNCTION VMNORM (N, V, W)                        00641700
CLLL. OPTIMIZE                                                          00641800
C------------------------------------------------------------------------00641900
C THIS FUNCTION ROUTINE COMPUTES THE WEIGHTED MAX-NORM                  00642000
C OF THE VECTOR OF LENGTH N CONTAINED IN THE ARRAY V, WITH WEIGHTS      00642100
C CONTAINED IN THE ARRAY W OF LENGTH N..                                00642200
```

```
C   VMNORM = MAX(I=1,...,N) ABS(V(I))*W(I)                                  00642300
C-------------------------------------------------------------------------00642400
      COMMON /RXNDTA/E(5),EXP(9,5),TEMPK,A(5),RK(5,2)                      00642500
     *        ,RCT(5,100,14),X(5),DELH(5),NCOEF(9,6),NRXTYP(5),NRXNTS      00642600
     *        ,NRXCID(30),IDIR(5),NRXNS,KEY,KEYPOS(5),NPOSPC(7,5)          00642700
     *        ,RXN(5,14)                                                   00642800
      CHARACTER RXN                                                       00642900
      INTEGER N,   I                                                      00643000
      DOUBLE PRECISION V, W,   VM                                         00643100
      DIMENSION V(N), W(N)                                                00643200
      VM = 0.0D0                                                          00643300
      DO 10 I = 1,NRXNTS                                                  00643400
 10     VM = DMAX1(VM,DABS(V(I))*W(I))                                    00643500
      VMNORM = VM                                                         00643600
      RETURN                                                              00643700
C---------------------- END OF FUNCTION VMNORM ----------------------------00643800
      END                                                                 00643900
      DOUBLE PRECISION FUNCTION BNORM (N, A, NRA, ML, MU, W)              00644000
CLLL. OPTIMIZE                                                            00644100
C-------------------------------------------------------------------------00644300
      INTEGER N, NRA, ML, MU                                              00644400
      INTEGER I, I1, JLO, JHI, J                                          00644500
      DOUBLE PRECISION A, W                                               00644600
      DOUBLE PRECISION AN, SUM                                            00644700
      DIMENSION A(NRA,N), W(N)                                            00644800
      AN = 0.0D0                                                          00644900
      DO 20 I = 1,N                                                       00645000
        SUM = 0.0D0                                                       00645100
        I1 = I + MU + 1                                                   00645200
        JLO = MAX0(I-ML,1)                                                00645300
        JHI = MIN0(I+MU,N)                                                00645400
        DO 10 J = JLO,JHI                                                 00645500
 10       SUM = SUM + DABS(A(I1-J,J))/W(J)                                00645600
        AN = DMAX1(AN,SUM*W(I))                                           00645700
 20     CONTINUE                                                          00645800
      BNORM = AN                                                          00645900
      RETURN                                                              00646000
C---------------------- END OF FUNCTION BNORM -----------------------------00646100
      END                                                                 00646200
      SUBROUTINE XERRWV (NUMERR, NERR, IERT, NI, I1, I2, NR,              00646300
     1                   R1, R2)                                          00646400
C                                                                         00646500
C   THIS ROUTINE HAS BEEN MODIFIED FOR USE ON THE FORTRAN 77 COMPILER     00646600
C   AT OKLAHOMA STATE UNIVERSITY. 11/85                                   00646700
C                                                                         00646800
      INTEGER  NMES, NERR, IERT, NI, I1, I2, NR,                          00646900
C   TOOK MSG OUT OF INTEGER STATEMENT                                     00647000
     1   I, LUN, LUNIT, MESFLG, NCPW, NCH, NWDS                           00647100
      DOUBLE PRECISION R1, R2                                             00647200
C     CHARACTER MSG                                                       00647300
C     DIMENSION MSG(NMES)                                                 00647400
      COMMON /EH0001/ MESFLG, LUNIT                                       00647500
      DATA NCPW/4/                                                        00647600
      IF (MESFLG .EQ. 0) GO TO 100                                        00647700
      LUN = LUNIT                                                         00647800
      NCH = MIN0(NMES,60)                                                 00647900
      NWDS = NCH/NCPW                                                     00648000
      IF (NCH .NE. NWDS*NCPW) NWDS = NWDS + 1                             00648100
      WRITE (LUN, 10) NUMERR                                              00648200
 10   FORMAT(/' **********   ERROR IN INTEGRATION ROUTINE  **********',   00648400
     1       /' ERROR CODE NUMBER IS',I3/)                                00648500
      IF (NI .EQ. 1) WRITE (LUN, 20) I1                                   00648600
 20   FORMAT(6X,23HIN ERROR MESSAGE,  I1 =,I10)                           00648700
      IF (NI .EQ. 2) WRITE (LUN, 30) I1,I2                                00648800
 30   FORMAT(6X,23HIN ERROR MESSAGE,  I1 =,I10,3X,,4HI2 =,I10)            00648900
      IF (NR .EQ. 1) WRITE (LUN, 40) R1                                   00649000
 40   FORMAT(6X,23HIN ERROR MESSAGE,  R1 =,D21.13)                        00649100
      IF (NR .EQ. 2) WRITE (LUN, 50) R1,R2                                00649200
 50   FORMAT(6X,15HIN ABOVE,  R1 =,D21.13,3X,4HR2 =,D21.13)               00649300
 100  IF (IERT .NE. 2) RETURN                                             00649400
      STOP                                                                00649500
C---------------------- END OF SUBROUTINE XERRWV ---------------------------00649600
```

```
      END                                                                00649700
      DOUBLE PRECISION FUNCTION FNORM (N, A, W)                          00649800
      INTEGER N,   I, J                                                  00649900
      DOUBLE PRECISION A,   W, AN, SUM                                   00650000
      DIMENSION A(N,N), W(N)                                             00650100
      AN = 0.0D0                                                         00650200
      DO 20 I = 1,N                                                      00650300
        SUM = 0.0D0                                                      00650400
        DO 10 J = 1,N                                                    00650500
 10       SUM = SUM + DABS(A(I,J))/W(J)                                  00650600
        AN = DMAX1(AN,SUM*W(I))                                          00650700
 20     CONTINUE                                                         00650800
      FNORM = AN                                                         00650900
      RETURN                                                             00651000
C---------------------- END OF FUNCTION FNORM -------------------------00651100
      END                                                                00651200
      SUBROUTINE DGEFA(A,LDA,N,IPVT,INFO)                                00651300
      INTEGER LDA,N,IPVT(1),INFO                                         00651400
      DOUBLE PRECISION A(LDA,1)                                          00651500
C                                                                        00651600
C146300                                                                  00651700
C146751                                                                  00651800
      DOUBLE PRECISION T                                                 00651900
      INTEGER IDAMAX,J,K,KP1,L,NM1                                       00652000
C                                                                        00652100
C                                                                        00652200
C     GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING                         00652300
C                                                                        00652400
      INFO = 0                                                           00652600
      NM1 = N - 1                                                        00652700
      IF (NM1 .LT. 1) GO TO 70                                           00652800
      DO 60 K = 1, NM1                                                   00652900
         KP1 = K + 1                                                     00653000
C                                                                        00653100
C        FIND L = PIVOT INDEX                                            00653200
C                                                                        00653300
         L = IDAMAX(N-K+1,A(K,K),1) + K - 1                              00653400
         IPVT(K) = L                                                     00653500
C                                                                        00653600
C        ZERO PIVOT IMPLIES THIS COLUMN ALREADY TRIANGULARIZED           00653700
C                                                                        00653800
         IF (A(L,K) .EQ. 0.0D0) GO TO 40                                 00653900
C                                                                        00654000
C           INTERCHANGE IF NECESSARY                                     00654100
C                                                                        00654200
            IF (L .EQ. K) GO TO 10                                       00654300
               T = A(L,K)                                                00654400
               A(L,K) = A(K,K)                                           00654500
               A(K,K) = T                                                00654600
   10       CONTINUE                                                     00654700
C                                                                        00654800
C           COMPUTE MULTIPLIERS                                          00654900
C                                                                        00655000
            T = -1.0D0/A(K,K)                                            00655100
            CALL DSCAL(N-K,T,A(K+1,K),1)                                 00655200
C                                                                        00655300
C           ROW ELIMINATION WITH COLUMN INDEXING                         00655400
C                                                                        00655500
            DO 30 J = KP1, N                                             00655600
               T = A(L,J)                                                00655700
               IF (L .EQ. K) GO TO 20                                    00655800
                  A(L,J) = A(K,J)                                        00655900
                  A(K,J) = T                                             00656000
   20          CONTINUE                                                  00656100
               CALL DAXPY(N-K,T,A(K+1,K),1,A(K+1,J),1)                   00656200
   30       CONTINUE                                                     00656300
         GO TO 50                                                        00656400
   40    CONTINUE                                                        00656500
            INFO = K                                                     00656600
   50    CONTINUE                                                        00656700
   60 CONTINUE                                                           00656800
   70 CONTINUE                                                           00656900
```

```
      IPVT(N) = N                                                       00657000
      IF (A(N,N) .EQ. 0.0D0) INFO = N                                   00657100
      RETURN                                                            00657200
      END                                                               00657300
C---  END DGEFA                                                         00657400
      SUBROUTINE DGESL(A,LDA,N,IPVT,B,JOB)                              00657600
      INTEGER LDA,N,IPVT(1),JOB                                         00657700
      DOUBLE PRECISION A(LDA,1),B(1)                                    00657800
      DOUBLE PRECISION DDOT,T                                           00658200
      INTEGER K,KB,L,NM1                                                00658300
      NM1 = N - 1                                                       00658600
      IF (JOB .NE. 0) GO TO 50                                          00658700
C                                                                       00658800
C        JOB = 0 , SOLVE  A * X = B                                     00658900
C        FIRST SOLVE  L*Y = B                                           00659000
C                                                                       00659100
         IF (NM1 .LT. 1) GO TO 30                                       00659200
         DO 20 K = 1, NM1                                               00659300
            L = IPVT(K)                                                 00659400
            T = B(L)                                                    00659500
            IF (L .EQ. K) GO TO 10                                      00659600
               B(L) = B(K)                                              00659700
               B(K) = T                                                 00659800
   10       CONTINUE                                                    00659900
            CALL DAXPY(N-K,T,A(K+1,K),1,B(K+1),1)                       00660000
   20    CONTINUE                                                       00660100
   30    CONTINUE                                                       00660200
C                                                                       00660300
C        NOW SOLVE  U*X = Y                                             00660400
C                                                                       00660500
         DO 40 KB = 1, N                                                00660600
            K = N + 1 - KB                                              00660700
            B(K) = B(K)/A(K,K)                                          00660800
            T = -B(K)                                                   00660900
            CALL DAXPY(K-1,T,A(1,K),1,B(1),1)                           00661000
   40    CONTINUE                                                       00661100
      GO TO 100                                                         00661200
   50 CONTINUE                                                          00661300
C                                                                       00661400
C        JOB = NONZERO, SOLVE  TRANS(A) * X = B                         00661500
C        FIRST SOLVE  TRANS(U)*Y = B                                    00661600
C                                                                       00661700
         DO 60 K = 1, N                                                 00661800
            T = DDOT(K-1,A(1,K),1,B(1),1)                               00661900
            B(K) = (B(K) - T)/A(K,K)                                    00662000
   60    CONTINUE                                                       00662100
C                                                                       00662200
C        NOW SOLVE TRANS(L)*X = Y                                       00662300
C                                                                       00662400
         IF (NM1 .LT. 1) GO TO 90                                       00662500
         DO 80 KB = 1, NM1                                              00662600
            K = N - KB                                                  00662700
            B(K) = B(K) + DDOT(N-K,A(K+1,K),1,B(K+1),1)                 00662800
            L = IPVT(K)                                                 00662900
            IF (L .EQ. K) GO TO 70                                      00663000
               T = B(L)                                                 00663100
               B(L) = B(K)                                              00663200
               B(K) = T                                                 00663300
   70       CONTINUE                                                    00663400
   80    CONTINUE                                                       00663500
   90    CONTINUE                                                       00663600
  100 CONTINUE                                                          00663700
      RETURN                                                            00663800
      END                                                               00663900
      SUBROUTINE DGBFA(ABD,LDA,N,ML,MU,IPVT,INFO)                       00664000
      INTEGER LDA,N,ML,MU,IPVT(1),INFO                                  00664100
      DOUBLE PRECISION ABD(LDA,1)                                       00664200
      DOUBLE PRECISION T                                                00664600
      INTEGER I,IDAMAX,IO,J,JU,JZ,JO,J1,K,KP1,L,LM,M,MM,NM1             00664700
      M = ML + MU + 1                                                   00665100
      INFO = 0                                                          00665200
C                                                                       00665300
```

```
C     ZERO INITIAL FILL-IN COLUMNS                                      00665400
C                                                                       00665500
      JO = MU + 2                                                       00665600
      J1 = MINO(N,M) - 1                                                00665700
      IF (J1 .LT. JO) GO TO 30                                          00665800
      DO 20 JZ = JO, J1                                                 00665900
         IO = M + 1 - JZ                                                00666000
         DO 10 I = IO, ML                                               00666100
            ABD(I,JZ) = 0.0D0                                           00666200
   10    CONTINUE                                                       00666300
   20 CONTINUE                                                          00666400
   30 CONTINUE                                                          00666500
      JZ = J1                                                           00666600
      JU = 0                                                            00666700
C                                                                       00666800
C     GAUSSIAN ELIMINATION WITH PARTIAL PIVOTING                        00666900
C                                                                       00667000
      NM1 = N - 1                                                       00667100
      IF (NM1 .LT. 1) GO TO 130                                         00667200
      DO 120 K = 1, NM1                                                 00667300
         KP1 = K + 1                                                    00667400
C                                                                       00667500
C        ZERO NEXT FILL-IN COLUMN                                       00667600
C                                                                       00667700
         JZ = JZ + 1                                                    00667800
         IF (JZ .GT. N) GO TO 50                                        00667900
         IF (ML .LT. 1) GO TO 50                                        00668000
            DO 40 I = 1, ML                                             00668100
               ABD(I,JZ) = 0.0D0                                        00668200
   40       CONTINUE                                                   00668300
   50    CONTINUE                                                       00668400
C                                                                       00668500
C        FIND L = PIVOT INDEX                                           00668600
C                                                                       00668700
         LM = MINO(ML,N-K)                                              00668800
         L = IDAMAX(LM+1,ABD(M,K),1) + M - 1                            00668900
         IPVT(K) = L + K - M                                            00669000
C                                                                       00669100
C        ZERO PIVOT IMPLIES THIS COLUMN ALREADY TRIANGULARIZED          00669200
C                                                                       00669300
         IF (ABD(L,K) .EQ. 0.0D0) GO TO 100                             00669400
C                                                                       00669500
C           INTERCHANGE IF NECESSARY                                    00669600
C                                                                       00669700
            IF (L .EQ. M) GO TO 60                                      00669800
               T = ABD(L,K)                                             00669900
               ABD(L,K) = ABD(M,K)                                      00670000
               ABD(M,K) = T                                             00670100
   60       CONTINUE                                                   00670200
C                                                                       00670300
C           COMPUTE MULTIPLIERS                                         00670400
C                                                                       00670500
            T = -1.0D0/ABD(M,K)                                         00670600
            CALL DSCAL(LM,T,ABD(M+1,K),1)                               00670700
C                                                                       00670800
C           ROW ELIMINATION WITH COLUMN INDEXING                        00670900
C                                                                       00671000
            JU = MINO(MAXO(JU,MU+IPVT(K)),N)                            00671100
            MM = M                                                      00671200
            IF (JU .LT. KP1) GO TO 90                                   00671300
            DO 80 J = KP1, JU                                           00671400
               L = L - 1                                                00671500
               MM = MM - 1                                              00671600
               T = ABD(L,J)                                             00671700
               IF (L .EQ. MM) GO TO 70                                  00671800
                  ABD(L,J) = ABD(MM,J)                                  00671900
                  ABD(MM,J) = T                                         00672000
   70          CONTINUE                                                 00672100
               CALL DAXPY(LM,T,ABD(M+1,K),1,ABD(MM+1,J),1)              00672200
   80       CONTINUE                                                   00672300
   90       CONTINUE                                                   00672400
         GO TO 110                                                      00672500
```

```
  100    CONTINUE                                                        00672600
            INFO = K                                                     00672700
  110    CONTINUE                                                        00672800
  120 CONTINUE                                                          00672900
  130 CONTINUE                                                          00673000
      IPVT(N) = N                                                       00673100
      IF (ABD(M,N) .EQ. 0.0D0) INFO = N                                 00673200
      RETURN                                                            00673300
      END                                                               00673400
      SUBROUTINE DGBSL(ABD,LDA,N,ML,MU,IPVT,B,JOB)                      00673600
      INTEGER LDA,N,ML,MU,IPVT(1),JOB                                   00673700
      DOUBLE PRECISION ABD(LDA,1),B(1)                                  00673800
      DOUBLE PRECISION DDOT,T                                           00674300
      INTEGER K,KB,L,LA,LB,LM,M,NM1                                     00674400
      M = MU + ML + 1                                                   00674700
      NM1 = N - 1                                                       00674800
      IF (JOB .NE. 0) GO TO 50                                          00674900
C                                                                       00675000
C        JOB = 0 , SOLVE  A * X = B                                     00675100
C        FIRST SOLVE L*Y = B                                            00675200
C                                                                       00675300
         IF (ML .EQ. 0) GO TO 30                                        00675400
         IF (NM1 .LT. 1) GO TO 30                                       00675500
            DO 20 K = 1, NM1                                            00675600
               LM = MIN0(ML,N-K)                                        00675700
               L = IPVT(K)                                              00675800
               T = B(L)                                                 00675900
               IF (L .EQ. K) GO TO 10                                   00676000
                  B(L) = B(K)                                           00676100
                  B(K) = T                                              00676200
   10          CONTINUE                                                 00676300
               CALL DAXPY(LM,T,ABD(M+1,K),1,B(K+1),1)                   00676400
   20       CONTINUE                                                    00676500
   30    CONTINUE                                                       00676600
C                                                                       00676700
C        NOW SOLVE  U*X = Y                                             00676800
C                                                                       00676900
         DO 40 KB = 1, N                                                00677000
            K = N + 1 - KB                                              00677100
            B(K) = B(K)/ABD(M,K)                                        00677200
            LM = MIN0(K,M) - 1                                          00677300
            LA = M - LM                                                 00677400
            LB = K - LM                                                 00677500
            T = -B(K)                                                   00677600
            CALL DAXPY(LM,T,ABD(LA,K),1,B(LB),1)                        00677700
   40    CONTINUE                                                       00677800
      GO TO 100                                                         00677900
   50 CONTINUE                                                          00678000
C                                                                       00678100
C        JOB = NONZERO, SOLVE  TRANS(A) * X = B                         00678200
C        FIRST SOLVE  TRANS(U)*Y = B                                    00678300
C                                                                       00678400
         DO 60 K = 1, N                                                 00678500
            LM = MIN0(K,M) - 1                                          00678600
            LA = M - LM                                                 00678700
            LB = K - LM                                                 00678800
            T = DDOT(LM,ABD(LA,K),1,B(LB),1)                            00678900
            B(K) = (B(K) - T)/ABD(M,K)                                  00679000
   60    CONTINUE                                                       00679100
C                                                                       00679200
C        NOW SOLVE TRANS(L)*X = Y                                       00679300
C                                                                       00679400
         IF (ML .EQ. 0) GO TO 90                                        00679500
         IF (NM1 .LT. 1) GO TO 90                                       00679600
            DO 80 KB = 1, NM1                                           00679700
               K = N - KB                                               00679800
               LM = MIN0(ML,N-K)                                        00679900
               B(K) = B(K) + DDOT(LM,ABD(M+1,K),1,B(K+1),1)             00680000
               L = IPVT(K)                                              00680100
               IF (L .EQ. K) GO TO 70                                   00680200
                  T = B(L)                                              00680300
                  B(L) = B(K)                                           00680400
```

```
                    B(K) = T                                                    00680500
   70             CONTINUE                                                     00680600
   80           CONTINUE                                                       00680700
   90       CONTINUE                                                           00680800
  100 CONTINUE                                                                00680900
      RETURN                                                                  00681000
      END                                                                     00681100
      SUBROUTINE DAXPY(N,DA,DX,INCX,DY,INCY)                                  00681300
C                                                                             00681400
C     CONSTANT TIMES A VECTOR PLUS A VECTOR.                                  00681500
C     USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO ONE.                        00681600
C     JACK DONGARRA, LINPACK, 3/11/78.                                        00681700
C                                                                             00681800
      DOUBLE PRECISION DX(1),DY(1),DA                                         00681900
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N                                       00682000
      IF(N.LE.0)RETURN                                                        00682300
      IF (DA .EQ. 0.0D0) RETURN                                               00682400
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20                                     00682500
C                                                                             00682600
C        CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS                      00682700
C          NOT EQUAL TO 1                                                     00682800
C                                                                             00682900
      IX = 1                                                                  00683000
      IY = 1                                                                  00683100
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1                                       00683200
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1                                       00683300
      DO 10 I = 1,N                                                           00683400
        DY(IY) = DY(IY) + DA*DX(IX)                                           00683500
        IX = IX + INCX                                                        00683600
        IY = IY + INCY                                                        00683700
   10 CONTINUE                                                                00683800
      RETURN                                                                  00683900
C                                                                             00684000
C        CODE FOR BOTH INCREMENTS EQUAL TO 1                                  00684100
C                                                                             00684200
C                                                                             00684300
C        CLEAN-UP LOOP                                                        00684400
C                                                                             00684500
   20 M = MOD(N,4)                                                            00684600
      IF( M .EQ. 0 ) GO TO 40                                                 00684700
      DO 30 I = 1,M                                                           00684800
        DY(I) = DY(I) + DA*DX(I)                                              00684900
   30 CONTINUE                                                                00685000
      IF( N .LT. 4 ) RETURN                                                   00685100
   40 MP1 = M + 1                                                             00685200
      DO 50 I = MP1,N,4                                                       00685300
        DY(I) = DY(I) + DA*DX(I)                                              00685400
        DY(I + 1) = DY(I + 1) + DA*DX(I + 1)                                  00685500
        DY(I + 2) = DY(I + 2) + DA*DX(I + 2)                                  00685600
        DY(I + 3) = DY(I + 3) + DA*DX(I + 3)                                  00685700
   50 CONTINUE                                                                00685800
      RETURN                                                                  00685900
      END                                                                     00686000
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++ 00686100
      DOUBLE PRECISION FUNCTION DDOT(N,DX,INCX,DY,INCY)                       00686200
C                                                                             00686300
C     FORMS THE DOT PRODUCT OF TWO VECTORS.                                   00686400
C     USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO ONE.                        00686500
C     JACK DONGARRA, LINPACK, 3/11/78.                                        00686600
C                                                                             00686700
      DOUBLE PRECISION DX(1),DY(1),DTEMP                                      00686800
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N                                       00686900
C                                                                             00687000
      DDOT = 0.0D0                                                            00687100
      DTEMP = 0.0D0                                                           00687200
      IF(N.LE.0)RETURN                                                        00687300
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20                                     00687400
C                                                                             00687500
C        CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS                      00687600
C          NOT EQUAL TO 1                                                     00687700
C                                                                             00687800
      IX = 1                                                                  00687900
```

```
      IY = 1                                                            00688000
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1                                 00688100
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1                                 00688200
      DO 10 I = 1,N                                                     00688300
        DTEMP = DTEMP + DX(IX)*DY(IY)                                   00688400
        IX = IX + INCX                                                  00688500
        IY = IY + INCY                                                  00688600
   10 CONTINUE                                                          00688700
      DDOT = DTEMP                                                      00688800
      RETURN                                                            00688900
C                                                                       00689000
C        CODE FOR BOTH INCREMENTS EQUAL TO 1                            00689100
C                                                                       00689200
C                                                                       00689300
C        CLEAN-UP LOOP                                                  00689400
C                                                                       00689500
   20 M = MOD(N,5)                                                      00689600
      IF( M .EQ. 0 ) GO TO 40                                           00689700
      DO 30 I = 1,M                                                     00689800
        DTEMP = DTEMP + DX(I)*DY(I)                                     00689900
   30 CONTINUE                                                          00690000
      IF( N .LT. 5 ) GO TO 60                                           00690100
   40 MP1 = M + 1                                                       00690200
      DO 50 I = MP1,N,5                                                 00690300
        DTEMP = DTEMP + DX(I)*DY(I) + DX(I + 1)*DY(I + 1) +             00690400
     *   DX(I + 2)*DY(I + 2) + DX(I + 3)*DY(I + 3) + DX(I + 4)*DY(I + 4)00690500
   50 CONTINUE                                                          00690600
   60 DDOT = DTEMP                                                      00690700
      RETURN                                                            00690800
      END                                                               00690900
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++00691000
      DOUBLE PRECISION FUNCTION D1MACH (IDUM)                           00691100
      INTEGER IDUM                                                      00691200
C-----------------------------------------------------------------------00691300
C THIS ROUTINE COMPUTES THE UNIT ROUNDOFF OF THE MACHINE IN DOUBLE      00691400
C PRECISION.  THIS IS DEFINED AS THE SMALLEST POSITIVE MACHINE NUMBER   00691500
C U SUCH THAT  1.0D0 + U .NE. 1.0D0 (IN DOUBLE PRECISION).              00691600
C-----------------------------------------------------------------------00691700
      DOUBLE PRECISION U, COMP                                          00691800
      U = 1.0D0                                                         00691900
 10   U = U*0.5D0                                                      00692000
      COMP = 1.0D0 + U                                                  00692100
      IF (COMP .NE. 1.0D0) GO TO 10                                     00692200
      D1MACH = U*2.0D0                                                  00692300
      RETURN                                                            00692400
C----------------------- END OF FUNCTION D1MACH ------------------------00692500
      END                                                               00692600
      INTEGER FUNCTION IDAMAX(N,DX,INCX)                                00692700
C                                                                       00692800
C     FINDS THE INDEX OF ELEMENT HAVING MAX. ABSOLUTE VALUE.            00692900
C     JACK DONGARRA, LINPACK, 3/11/78.                                  00693000
C                                                                       00693100
      DOUBLE PRECISION DX(1),DMAX                                       00693200
      INTEGER I,INCX,IX,N                                               00693300
C                                                                       00693400
      IDAMAX = 0                                                        00693500
      IF( N .LT. 1 ) RETURN                                             00693600
      IDAMAX = 1                                                        00693700
      IF(N.EQ.1)RETURN                                                  00693800
      IF(INCX.EQ.1)GO TO 20                                             00693900
C                                                                       00694000
C        CODE FOR INCREMENT NOT EQUAL TO 1                              00694100
C                                                                       00694200
      IX = 1                                                            00694300
      DMAX = DABS(DX(1))                                                00694400
      IX = IX + INCX                                                    00694500
      DO 10 I = 2,N                                                     00694600
         IF(DABS(DX(IX)).LE.DMAX) GO TO 5                               00694700
         IDAMAX = I                                                     00694800
         DMAX = DABS(DX(IX))                                            00694900
    5    IX = IX + INCX                                                 00695000
   10 CONTINUE                                                          00695100
```

```
      RETURN                                                            00695200
C                                                                       00695300
C        CODE FOR INCREMENT EQUAL TO 1                                  00695400
C                                                                       00695500
   20 DMAX = DABS(DX(1))                                                00695600
      DO 30 I = 2,N                                                     00695700
         IF(DABS(DX(I)).LE.DMAX) GO TO 30                               00695800
         IDAMAX = I                                                     00695900
         DMAX = DABS(DX(I))                                             00696000
   30 CONTINUE                                                          00696100
      RETURN                                                            00696200
      END                                                               00696300
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++00696400
C     BLOCK DATA                                                        00696500
C     INTEGER ILLIN, IDUMA, NTREP, IDUMB, IOWNS, ICOMM, MESFLG, LUNIT   00696600
C     DOUBLE PRECISION ROWND, ROWNS, RCOMM                              00696700
C     COMMON /LS0001/ ROWND, ROWNS(209), RCOMM(9),                      00696800
C    1   ILLIN, IDUMA(10), NTREP, IDUMB(2), IOWNS(6), ICOMM(19)         00696900
C     COMMON /EH0001/ MESFLG, LUNIT                                     00697000
C     DATA ILLIN/0/, NTREP/0/                                           00697100
C     DATA MESFLG/1/, LUNIT/6/                                          00697200
C                                                                       00697300
C----------------------- END OF BLOCK DATA -----------------------------00697400
C     END                                                               00697500
      SUBROUTINE SVCMA (RSAV, ISAV)                                     00697600
      INTEGER ISAV                                                      00697700
      INTEGER IEH, ILS, ILSA                                            00697800
      INTEGER I, LENRLS, LENILS, LENRLA, LENILA                         00697900
      DOUBLE PRECISION RSAV                                             00698000
      DOUBLE PRECISION RLS, RLSA                                        00698100
      DIMENSION RSAV(1), ISAV(1)                                        00698200
      COMMON /LS0001/ RLS(219), ILS(39)                                 00698300
      COMMON /LSA001/ RLSA(22), ILSA(9)                                 00698400
      COMMON /EH0001/ IEH(2)                                            00698500
      DATA LENRLS/219/, LENILS/39/, LENRLA/22/, LENILA/9/               00698600
C                                                                       00698700
      DO 10 I = 1,LENRLS                                                00698900
 10     RSAV(I) = RLS(I)                                                00699000
      DO 15 I = 1,LENRLA                                                00699100
 15     RSAV(LENRLS+I) = RLSA(I)                                        00699200
C                                                                       00699300
      DO 20 I = 1,LENILS                                                00699400
 20     ISAV(I) = ILS(I)                                                00699500
      DO 25 I = 1,LENILA                                                00699600
 25     ISAV(LENILS+I) = ILSA(I)                                        00699700
C                                                                       00699800
      ISAV(LENILS+LENILA+1) = IEH(1)                                    00699900
      ISAV(LENILS+LENILA+2) = IEH(2)                                    00700000
      RETURN                                                            00700100
C----------------------- END OF SUBROUTINE SVCMA -----------------------00700200
      END                                                               00700300
C75771                                                                  00700400
      SUBROUTINE RCHEK (JOB, G, NEQ, Y, YH, NYH, GO, G1, GX,            00700500
     1           JROOT,IRT,INDX)                                        00700600
CLLL. OPTIMIZE                                                          00700700
      EXTERNAL G                                                        00700800
      INTEGER JOB, NEQ, NYH, JROOT, IRT                                 00700900
      DOUBLE PRECISION Y, YH, GO, G1, GX                                00701000
      DIMENSION Y(1), YH(NYH,1), GO(1), G1(1), GX(1), JROOT(1)          00701100
CN                                                                      00701200
      INTEGER IOWND, IOWNS,                                             00701300
     1   ICF, IERPJ, IERSL, JCUR, JSTART, KFLAG, L, METH, MITER,        00701400
     2   MAXORD, MAXCOR, MSBP, MXNCF, N, NQ, NST, NFE, NJE, NQU         00701500
      INTEGER IOWND3, IOWNR3, IRFND, ITASKC, NGC, NGE                   00701600
      INTEGER I, IFLAG, JFLAG                                           00701700
      DOUBLE PRECISION ROWND, ROWNS,                                    00701800
     1   CCMAX, ELO, H, HMIN, HMXI, HU, RC, TN, UROUND                  00701900
      DOUBLE PRECISION ROWNR3, TO, TLAST, TOUTC                         00702000
      DOUBLE PRECISION HMING, T1, TEMP1, TEMP2, X                       00702100
      LOGICAL ZROOT                                                     00702200
      COMMON /LS0001/ ROWND, ROWNS(209),                                00702300
     2   CCMAX, ELO, H, HMIN, HMXI, HU, RC, TN, UROUND,                 00702400
```

```
     3   IOWND(14), IOWNS(6),                                              00702500
     4   ICF, IERPJ, IERSL, JCUR, JSTART, KFLAG, L, METH, MITER,           00702600
     5   MAXORD, MAXCOR, MSBP, MXNCF, N, NQ, NST, NFE, NJE, NQU            00702700
      COMMON /LSR001/ ROWNR3(2), TO, TLAST, TOUTC,                         00702800
     1   IOWND3(3), IOWNR3(2), IRFND, ITASKC, NGC, NGE                     00702900
C-----------------------------------------------------------------------00703000
C94601                                                                    00703100
C94941                                                                    00703200
C-----------------------------------------------------------------------00703300
      IRT = 0                                                             00703600
      DO 10 I = 1,NGC                                                     00703700
 10     JROOT(I) = 0                                                     00703800
      HMING = (DABS(TN) + DABS(H))*UROUND*100.0D0                         00703900
C                                                                         00704000
      GO TO (100, 200, 300), JOB                                          00704100
C                                                                         00704200
C EVALUATE G AT INITIAL T, AND CHECK FOR ZERO VALUES. ------------------00704300
 100  CONTINUE                                                            00704400
      T0 = TN                                                             00704500
      CALL G (NEQ, T0, Y, NGC, G0,INDX)                                   00704600
C----------  RCHEK                                                        00704700
      NGE = 1                                                             00704800
      ZROOT = .FALSE.                                                     00704900
      DO 110 I = 1,NGC                                                    00705000
 110    IF (DABS(G0(I)) .LE. 0.0D0) ZROOT = .TRUE.                       00705100
      IF (.NOT. ZROOT) GO TO 190                                          00705200
C G HAS A ZERO AT T.  LOOK AT G AT T + (SMALL INCREMENT). --------------00705300
      TEMP1 = DSIGN(HMING,H)                                              00705400
      T0 = T0 + TEMP1                                                     00705500
      TEMP2 = TEMP1/H                                                     00705600
      DO 120 I = 1,N                                                      00705700
 120    Y(I) = Y(I) + TEMP2*YH(I,2)                                      00705800
      CALL G (NEQ, T0, Y, NGC, G0,INDX)                                   00705900
      NGE = NGE + 1                                                       00706000
      ZROOT = .FALSE.                                                     00706100
      DO 130 I = 1,NGC                                                    00706200
 130    IF (DABS(G0(I)) .LE. 0.0D0) ZROOT = .TRUE.                       00706300
      IF (.NOT. ZROOT) GO TO 190                                          00706400
C G HAS A ZERO AT T AND ALSO CLOSE TO T.  TAKE ERROR RETURN. -----------00706500
      IRT = -1                                                            00706600
      RETURN                                                              00706700
C                                                                         00706800
 190  CONTINUE                                                            00706900
      RETURN                                                              00707000
C                                                                         00707100
C                                                                         00707200
 200  CONTINUE                                                            00707300
      IF (IRFND .EQ. 0) GO TO 260                                         00707400
C IF A ROOT WAS FOUND ON THE PREVIOUS STEP, EVALUATE G0 = G(T0). -------00707500
      CALL INTDY (T0, 0, YH, NYH, Y, IFLAG)                               00707600
      CALL G (NEQ, T0, Y, NGC, G0,INDX)                                   00707700
C------------   RCHEK                                                    00707800
      NGE = NGE + 1                                                       00707900
      ZROOT = .FALSE.                                                     00708000
      DO 210 I = 1,NGC                                                    00708100
 210    IF (DABS(G0(I)) .LE. 0.0D0) ZROOT = .TRUE.                       00708200
      IF (.NOT. ZROOT) GO TO 260                                          00708300
C G HAS A ZERO AT T0.  LOOK AT G AT T + (SMALL INCREMENT). -------------00708400
      TEMP1 = DSIGN(HMING,H)                                              00708500
      T0 = T0 + TEMP1                                                     00708600
      IF ((T0 - TN)*H .LT. 0.0D0) GO TO 230                               00708700
      TEMP2 = TEMP1/H                                                     00708800
      DO 220 I = 1,N                                                      00708900
 220    Y(I) = Y(I) + TEMP2*YH(I,2)                                      00709000
      GO TO 240                                                           00709100
 230  CALL INTDY (T0, 0, YH, NYH, Y, IFLAG)                               00709200
 240  CALL G (NEQ, T0, Y, NGC, G0,INDX)                                   00709300
C------------   RCHEK                                                    00709400
      NGE = NGE + 1                                                       00709500
      ZROOT = .FALSE.                                                     00709600
      DO 250 I = 1,NGC                                                    00709700
        IF (DABS(G0(I)) .GT. 0.0D0) GO TO 250                             00709800
```

```
        JROOT(I) = 1                                                    00709900
        ZROOT = .TRUE.                                                  00710000
 250    CONTINUE                                                        00710100
      IF (.NOT. ZROOT) GO TO 260                                        00710200
C G HAS A ZERO AT TO AND ALSO CLOSE TO TO.  RETURN ROOT. ---------------00710300
      IRT = 1                                                           00710400
      RETURN                                                            00710500
C     HERE, GO DOES NOT HAVE A ROOT                                     00710600
C GO HAS NO ZERO COMPONENTS.  PROCEED TO CHECK RELEVANT INTERVAL. ------00710700
 260  IF (TN .EQ. TLAST) GO TO 390                                      00710800
C                                                                       00710900
 300  CONTINUE                                                          00711000
C SET T1 TO TN OR TOUTC, WHICHEVER COMES FIRST, AND GET G AT T1. -------00711100
      IF (ITASKC.EQ.2 .OR. ITASKC.EQ.3 .OR. ITASKC.EQ.5) GO TO 310      00711200
      IF ((TOUTC - TN)*H .GE. 0.0D0) GO TO 310                          00711300
      T1 = TOUTC                                                        00711400
      IF ((T1 - TO)*H .LE. 0.0D0) GO TO 390                             00711500
      CALL INTDY (T1, 0, YH, NYH, Y, IFLAG)                             00711600
      GO TO 330                                                         00711700
 310  T1 = TN                                                           00711800
      DO 320 I = 1,N                                                    00711900
 320    Y(I) = YH(I,1)                                                  00712000
 330  CALL G (NEQ, T1, Y, NGC, G1,INDX)                                 00712100
C--------------  RCHEK                                                 00712200
      NGE = NGE + 1                                                     00712300
C CALL ROOTS TO SEARCH FOR ROOT IN INTERVAL FROM TO TO T1. -------------00712400
      JFLAG = 0                                                         00712500
 350  CONTINUE                                                          00712600
      CALL ROOTS (NGC, HMING, JFLAG, TO, T1, GO, G1, GX, X, JROOT)      00712700
      IF (JFLAG .GT. 1) GO TO 360                                       00712800
      CALL INTDY (X, 0, YH, NYH, Y, IFLAG)                              00712900
      CALL G (NEQ, X, Y, NGC, GX,INDX)                                  00713000
      NGE = NGE + 1                                                     00713100
      GO TO 350                                                         00713200
 360  TO = X                                                            00713300
      CALL DCOPY (NGC, GX, 1, GO, 1)                                    00713400
      IF (JFLAG .EQ. 4) GO TO 390                                       00713500
C FOUND A ROOT.  INTERPOLATE TO X AND RETURN. --------------------------00713600
      CALL INTDY (X, 0, YH, NYH, Y, IFLAG)                              00713700
      IRT = 1                                                           00713800
      RETURN                                                            00713900
C                                                                       00714000
 390  CONTINUE                                                          00714100
      RETURN                                                            00714200
C----------------------- END OF SUBROUTINE RCHEK ----------------------00714300
      END                                                               00714400
      SUBROUTINE ROOTS (NG, HMIN, JFLAG, XO, X1, GO, G1, GX, X, JROOT)  00714600
CLLL. OPTIMIZE                                                          00714700
      INTEGER NG, JFLAG, JROOT                                          00714800
      DOUBLE PRECISION HMIN, XO, X1, GO, G1, GX, X                      00714900
      DIMENSION GO(NG), G1(NG), GX(NG), JROOT(NG)                       00715000
      INTEGER IOWND3, IMAX, LAST, IDUM3                                 00715100
      DOUBLE PRECISION ALPHA, X2, RDUM3                                 00715200
      COMMON /LSR001/ ALPHA, X2, RDUM3(3),                              00715300
     1   IOWND3(3), IMAX, LAST, IDUM3(4)                                00715400
      INTEGER I, IMXOLD, NXLAST                                         00715900
      DOUBLE PRECISION T2, TMAX, ZERO                                   00716000
      LOGICAL ZROOT, SGNCHG, XROOT                                      00716100
      DATA ZERO/0.0D0/                                                  00716200
      IF (JFLAG .EQ. 1) GO TO 200                                       00716500
C JFLAG .NE. 1.  CHECK FOR CHANGE IN SIGN OF G OR ZERO AT X1. ----------00716600
      IMAX = 0                                                          00716700
      TMAX = ZERO                                                       00716800
      ZROOT = .FALSE.                                                   00716900
      DO 120 I = 1,NG                                                   00717000
        IF (DABS(G1(I)) .GT. ZERO) GO TO 110                            00717100
        ZROOT = .TRUE.                                                  00717200
        GO TO 120                                                       00717300
C AT THIS POINT, GO(I) HAS BEEN CHECKED AND CANNOT BE ZERO. ------------00717400
 110    IF (DSIGN(1.0D0,GO(I)) .EQ. DSIGN(1.0D0,G1(I))) GO TO 120       00717500
          T2 = DABS(G1(I)/(G1(I)-GO(I)))                                00717600
          IF (T2 .LE. TMAX) GO TO 120                                   00717700
```

```
            TMAX = T2                                                   00717800
            IMAX = I                                                    00717900
 120      CONTINUE                                                      00718000
      IF (IMAX .GT. 0) GO TO 130                                        00718100
      SGNCHG = .FALSE.                                                  00718200
      GO TO 140                                                         00718300
 130  SGNCHG = .TRUE.                                                   00718400
 140  IF (.NOT. SGNCHG) GO TO 400                                       00718500
C THERE IS A SIGN CHANGE.  FIND THE FIRST ROOT IN THE INTERVAL. --------00718600
      XROOT = .FALSE.                                                   00718700
      NXLAST = 0                                                        00718800
      LAST = 1                                                          00718900
C                                                                       00719000
C REPEAT UNTIL THE FIRST ROOT IN THE INTERVAL IS FOUND.  LOOP POINT. ---00719100
 150  CONTINUE                                                          00719200
      IF (XROOT) GO TO 300                                              00719300
      IF (NXLAST .EQ. LAST) GO TO 160                                   00719400
      ALPHA = 1.0D0                                                     00719500
      GO TO 180                                                         00719600
 160  IF (LAST .EQ. 0) GO TO 170                                        00719700
      ALPHA = 0.5D0*ALPHA                                               00719800
      GO TO 180                                                         00719900
 170  ALPHA = 2.0D0*ALPHA                                               00720000
 180  X2 = X1 - (X1-X0)*G1(IMAX)/(G1(IMAX) - ALPHA*G0(IMAX))            00720100
      IF ((DABS(X2-X0) .LT. HMIN) .AND.                                 00720200
     1   (DABS(X1-X0) .GT. 10.0D0*HMIN)) X2 = X0 + 0.1D0*(X1-X0)        00720300
      JFLAG = 1                                                         00720400
      X = X2                                                            00720500
C RETURN TO THE CALLING ROUTINE TO GET A VALUE OF GX = G(X). ----------00720600
      RETURN                                                            00720700
C CHECK TO SEE IN WHICH INTERVAL G CHANGES SIGN. ----------------------00720800
 200  IMXOLD = IMAX                                                     00720900
      IMAX = 0                                                          00721000
      TMAX = ZERO                                                       00721100
      ZROOT = .FALSE.                                                   00721200
      DO 220 I = 1,NG                                                   00721300
        IF (DABS(GX(I)) .GT. ZERO) GO TO 210                            00721400
        ZROOT = .TRUE.                                                  00721500
        GO TO 220                                                       00721600
C NEITHER G0(I) NOR GX(I) CAN BE ZERO AT THIS POINT. ------------------00721700
 210    IF (DSIGN(1.0D0,G0(I)) .EQ. DSIGN(1.0D0,GX(I))) GO TO 220       00721800
          T2 = DABS(GX(I)/(GX(I) - G0(I)))                              00721900
          IF (T2 .LE. TMAX) GO TO 220                                   00722000
            TMAX = T2                                                   00722100
            IMAX = I                                                    00722200
 220    CONTINUE                                                        00722300
      IF (IMAX .GT. 0) GO TO 230                                        00722400
      SGNCHG = .FALSE.                                                  00722500
      IMAX = IMXOLD                                                     00722600
      GO TO 240                                                         00722700
 230  SGNCHG = .TRUE.                                                   00722800
 240  NXLAST = LAST                                                     00722900
      IF (.NOT. SGNCHG) GO TO 250                                       00723000
C SIGN CHANGE BETWEEN X0 AND X2, SO REPLACE X1 WITH X2. ----------------00723100
      X1 = X2                                                           00723200
      CALL DCOPY (NG, GX, 1, G1, 1)                                     00723300
      LAST = 1                                                          00723400
      XROOT = .FALSE.                                                   00723500
      GO TO 270                                                         00723600
 250  IF (.NOT. ZROOT) GO TO 260                                        00723700
C ZERO VALUE AT X2 AND NO SIGN CHANGE IN (X0,X2), SO X2 IS A ROOT. -----00723800
      X1 = X2                                                           00723900
      CALL DCOPY (NG, GX, 1, G1, 1)                                     00724000
      XROOT = .TRUE.                                                    00724100
      GO TO 270                                                         00724200
C NO SIGN CHANGE BETWEEN X0 AND X2.  REPLACE X0 WITH X2. ---------------00724300
 260  CONTINUE                                                          00724400
      CALL DCOPY (NG, GX, 1, G0, 1)                                     00724500
      X0 = X2                                                           00724600
      LAST = 0                                                          00724700
      XROOT = .FALSE.                                                   00724800
 270  IF (DABS(X1-X0) .LE. HMIN) XROOT = .TRUE.                         00724900
```

```
      GO TO 150                                                          00725000
C                                                                        00725100
C RETURN WITH X1 AS THE ROOT.  SET JROOT.  SET X = X1 AND GX = G1. -----00725200
 300  JFLAG = 2                                                          00725300
      X = X1                                                             00725400
      CALL DCOPY (NG, G1, 1, GX, 1)                                      00725500
      DO 320 I = 1,NG                                                    00725600
        JROOT(I) = 0                                                     00725700
        IF (DABS(G1(I)) .GT. ZERO) GO TO 310                             00725800
          JROOT(I) = 1                                                   00725900
          GO TO 320                                                      00726000
 310    IF (DSIGN(1.0D0,G0(I)) .NE. DSIGN(1.0D0,G1(I))) JROOT(I) = 1     00726100
 320    CONTINUE                                                         00726200
      RETURN                                                             00726300
C                                                                        00726400
C NO SIGN CHANGE IN THE INTERVAL.  CHECK FOR ZERO AT RIGHT ENDPOINT. ---00726500
 400  IF (.NOT. ZROOT) GO TO 420                                         00726600
C                                                                        00726700
C ZERO VALUE AT X1 AND NO SIGN CHANGE IN (X0,X1).  RETURN JFLAG = 3. ---00726800
      X = X1                                                             00726900
      CALL DCOPY (NG, G1, 1, GX, 1)                                      00727000
      DO 410 I = 1,NG                                                    00727100
        JROOT(I) = 0                                                     00727200
        IF (DABS(G1(I)) .LE. ZERO) JROOT (I) = 1                         00727300
 410  CONTINUE                                                           00727400
      JFLAG = 3                                                          00727500
      RETURN                                                             00727600
C                                                                        00727700
C NO SIGN CHANGES IN THIS INTERVAL.  SET X = X1, RETURN JFLAG = 4. -----00727800
 420  CALL DCOPY (NG, G1, 1, GX, 1)                                      00727900
      X = X1                                                             00728000
      JFLAG = 4                                                          00728100
      RETURN                                                             00728200
C----------------------- END OF SUBROUTINE ROOTS -----------------------00728300
      END                                                                00728400
      SUBROUTINE SVCAR (RSAV, ISAV)                                      00728500
C----------------------------------------------------------------------00728600
C THIS ROUTINE STORES IN RSAV AND ISAV THE CONTENTS.OF COMMON BLOCKS     00728700
C LS0001, LSA001, LSR001, AND EH0001, WHICH ARE USED INTERNALLY IN THE   00728800
C LSODAR PACKAGE.                                                        00728900
C                                                                        00729000
C RSAV = REAL ARRAY OF LENGTH 246 OR MORE.                               00729100
C ISAV = INTEGER ARRAY OF LENGTH 59 OR MORE.                             00729200
C----------------------------------------------------------------------00729300
      INTEGER ISAV                                                       00729400
      INTEGER IEH, ILS, ILSA, ILSR                                       00729500
      INTEGER I, IOFF, LENRLS, LENILS, LENRLA, LENILA, LENRLR, LENILR    00729600
      DOUBLE PRECISION RSAV                                              00729700
      DOUBLE PRECISION RLS, RLSA, RLSR                                   00729800
      DIMENSION RSAV(1), ISAV(1)                                         00729900
      COMMON /LS0001/ RLS(219), ILS(39)                                  00730000
      COMMON /LSA001/ RLSA(22), ILSA(9)                                  00730100
      COMMON /LSR001/ RLSR(5), ILSR(9)                                   00730200
      COMMON /EH0001/ IEH(2)                                             00730300
      DATA LENRLS/219/, LENILS/39/, LENRLA/22/, LENILA/9/                00730400
      DATA LENRLR/5/, LENILR/9/                                          00730500
      DO 10 I = 1,LENRLS                                                 00730800
 10     RSAV(I) = RLS(I)                                                 00730900
      DO 15 I = 1,LENRLA                                                 00731000
 15     RSAV(LENRLS+I) = RLSA(I)                                         00731100
      IOFF = LENRLS + LENRLA                                             00731200
      DO 20 I = 1,LENRLR                                                 00731300
 20     RSAV(IOFF+I) = RLSR(I)                                           00731400
C                                                                        00731500
      DO 30 I = 1,LENILS                                                 00731600
 30     ISAV(I) = ILS(I)                                                 00731700
      DO 35 I = 1,LENILA                                                 00731800
 35     ISAV(LENILS+I) = ILSA(I)                                         00731900
      IOFF = LENILS + LENILA                                             00732000
      DO 40 I = 1,LENILR                                                 00732100
 40     ISAV(IOFF+I) = ILSR(I)                                           00732200
C                                                                        00732300
```

```
      IOFF = IOFF + LENILR                                              00732400
      ISAV(IOFF+1) = IEH(1)                                             00732500
      ISAV(IOFF+2) = IEH(2)                                             00732600
      RETURN                                                            00732700
C---------------------- END OF SUBROUTINE SVCAR ----------------------00732800
      END                                                               00732900
      SUBROUTINE RSCAR (RSAV, ISAV)                                     00733100
C-----------------------------------------------------------------------00733200
C THIS ROUTINE RESTORES FROM RSAV AND ISAV THE CONTENTS OF COMMON       00733300
C BLOCKS LS0001, LSA001, LSR001, AND EH0001, WHICH ARE USED INTERNALLY  00733400
C IN THE LSODAR PACKAGE.  THIS PRESUMES THAT RSAV AND ISAV WERE         00733500
C LOADED BY MEANS OF SUBROUTINE SVCAR OR THE EQUIVALENT.                00733600
C-----------------------------------------------------------------------00733700
      INTEGER ISAV                                                      00733800
      INTEGER IEH, ILS, ILSA, ILSR                                      00733900
      INTEGER I, IOFF, LENRLS, LENILS, LENRLA, LENILA, LENRLR, LENILR   00734000
      DOUBLE PRECISION RSAV                                             00734100
      DOUBLE PRECISION RLS, RLSA, RLSR                                  00734200
      DIMENSION RSAV(1), ISAV(1)                                        00734300
      COMMON /LS0001/ RLS(219), ILS(39)                                 00734400
      COMMON /LSA001/ RLSA(22), ILSA(9)                                 00734500
      COMMON /LSR001/ RLSR(5), ILSR(9)                                  00734600
      COMMON /EH0001/ IEH(2)                                            00734700
      DATA LENRLS/219/, LENILS/39/, LENRLA/22/, LENILA/9/               00734800
      DATA LENRLR/5/, LENILR/9/                                         00734900
      DO 10 I = 1,LENRLS                                                00735200
 10     RLS(I) = RSAV(I)                                                00735300
      DO 15 I = 1,LENRLA                                                00735400
 15     RLSA(I) = RSAV(LENRLS+I)                                        00735500
      IOFF = LENRLS + LENRLA                                            00735600
      DO 20 I = 1,LENRLR                                                00735700
 20     RLSR(I) = RSAV(IOFF+I)                                          00735800
C                                                                       00735900
      DO 30 I = 1,LENILS                                                00736000
 30     ILS(I) = ISAV(I)                                                00736100
      DO 35 I = 1,LENILA                                                00736200
 35     ILSA(I) = ISAV(LENILS+I)                                        00736300
      IOFF = LENILS + LENILA                                            00736400
      DO 40 I = 1,LENILR                                                00736500
 40     ILSR(I) = ISAV(IOFF+I)                                          00736600
C                                                                       00736700
      IOFF = IOFF + LENILR                                              00736800
      IEH(1) = ISAV(IOFF+1)                                             00736900
      IEH(2) = ISAV(IOFF+2)                                             00737000
      RETURN                                                            00737100
C---------------------- END OF SUBROUTINE RSCAR ----------------------00737200
      END                                                               00737300
      SUBROUTINE XSETF (MFLAG)                                          00737400
C                                                                       00737500
C THIS ROUTINE RESETS THE PRINT CONTROL FLAG MFLAG.                     00737600
C                                                                       00737700
      INTEGER MFLAG, MESFLG, LUNIT                                      00737800
      COMMON /EH0001/ MESFLG, LUNIT                                     00737900
      IF (MFLAG .EQ. 0 .OR. MFLAG .EQ. 1) MESFLG = MFLAG                00738200
      RETURN                                                            00738300
C---------------------- END OF SUBROUTINE XSETF ----------------------00738400
      END                                                               00738500
      SUBROUTINE XSETUN (LUN)                                           00738600
C                                                                       00738700
C THIS ROUTINE RESETS THE LOGICAL UNIT NUMBER FOR MESSAGES.             00738800
C                                                                       00738900
      INTEGER LUN, MESFLG, LUNIT                                        00739000
      COMMON /EH0001/ MESFLG, LUNIT                                     00739100
C                                                                       00739200
C *** MATHLIB USES NEGATIVE UNITS...SO WE HAVE COMMENTED THIS LINE OUT  00739300
C     IF (LUN .GT. 0) LUNIT = LUN                                       00739400
C AND ADDED OUR OWN LINE  "LUNIT = LUN"                                 00739500
      LUNIT = LUN                                                       00739700
      RETURN                                                            00739800
C---------------------- END OF SUBROUTINE XSETUN ----------------------00739900
      END                                                               00740000
      SUBROUTINE DCOPY(N,SX,INCX,SY,INCY)                               00740200
```

```
C                                                                       00740300
C     COPIES A VECTOR, X, TO A VECTOR, Y.                               00740400
C     USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO 1.                    00740500
C     JACK DONGARRA, LINPACK, 3/11/78.                                  00740600
C                                                                       00740700
      DOUBLE PRECISION SX(1),SY(1)                                      00740800
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N                                 00740900
      IF(N.LE.0)RETURN                                                  00741200
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20                               00741300
C                                                                       00741400
C        CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS                00741500
C          NOT EQUAL TO 1                                               00741600
C                                                                       00741700
      IX = 1                                                            00741800
      IY = 1                                                            00741900
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1                                 00742000
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1                                 00742100
      DO 10 I = 1,N                                                     00742200
        SY(IY) = SX(IX)                                                 00742300
        IX = IX + INCX                                                  00742400
        IY = IY + INCY                                                  00742500
   10 CONTINUE                                                          00742600
      RETURN                                                            00742700
C                                                                       00742800
C        CODE FOR BOTH INCREMENTS EQUAL TO 1                            00742900
C                                                                       00743000
C                                                                       00743100
C        CLEAN-UP LOOP                                                  00743200
C                                                                       00743300
   20 M = MOD(N,7)                                                      00743400
      IF( M .EQ. 0 ) GO TO 40                                           00743500
      DO 30 I = 1,M                                                     00743600
        SY(I) = SX(I)                                                   00743700
   30 CONTINUE                                                          00743800
      IF( N .LT. 7 ) RETURN                                             00743900
   40 MP1 = M + 1                                                       00744000
      DO 50 I = MP1,N,7                                                 00744100
        SY(I) = SX(I)                                                   00744200
        SY(I + 1) = SX(I + 1)                                           00744300
        SY(I + 2) = SX(I + 2)                                           00744400
        SY(I + 3) = SX(I + 3)                                           00744500
        SY(I + 4) = SX(I + 4)                                           00744600
        SY(I + 5) = SX(I + 5)                                           00744700
        SY(I + 6) = SX(I + 6)                                           00744800
   50 CONTINUE                                                          00744900
      RETURN                                                            00745000
      END                                                               00745100
C----------------------   END SUBROUTINE DCOPY   -------------------    00745200
      BLOCK DATA                                                        00745300
      INTEGER ILLIN, IDUMA, NTREP, IDUMB, IOWNS, ICOMM, MESFLG, LUNIT   00745400
      DOUBLE PRECISION ROWND, ROWNS, RCOMM                              00745500
      COMMON /LS0001/ ROWND, ROWNS(209), RCOMM(9),                      00745600
     1   ILLIN, IDUMA(10), NTREP, IDUMB(2), IOWNS(6), ICOMM(19)         00745700
      COMMON /EH0001/ MESFLG, LUNIT                                     00745800
      DATA ILLIN/0/, NTREP/0/                                           00745900
      DATA MESFLG/1/, LUNIT/6/                                          00746000
C---------------------- END OF BLOCK DATA ----------------------------00746200
      END                                                               00746400
//LKED.SYSLMOD DD DSNAME=U14319A.PROCSIM.LOAD,DISP=SHR,                 00746410
//LKED.SYSIN DD *                                                       00746420
   NAME KINETX(R)                                                       00746430
//                                                                      00746700
```

VITA

William Joseph Vedder Jr.

Candidate for the Degree of

Master of Science

Thesis: CHEMICAL PROCESS SIMULATOR WITH PLUG FLOW REACTOR

Major Field: Chemical Engineering

Biographical:

Personal Data: Born in Evergreen Park, Illinois, January 5, 1955, the son of Mr. and Mrs. William Vedder.

Education: Graduated from Brother Rice High School, Chicago, Illinois, in May 1973; received the Bachelor of Science degree in Chemical Engineering from Oklahoma State University in May, 1984; completed requirements for the Master of Science degree in December, 1985.

Professional Experience: Employed as a Research Assistant at Oklahoma State University during the spring and summer semesters of 1984; employed as a Teaching Assistant at Oklahoma State University during the fall semester of 1984 and spring semester of 1985; during the summer of 1985, employed as a Research Assistant at Oklahoma State University; presently employed as a Teaching Assistant at Oklahoma State University.