187
12-26-78

MASTER

DR 2059

UCRL-52561

DYNSYL: A GENERAL-PURPOSE DYNAMIC SIMULATOR FOR CHEMICAL PROCESSES

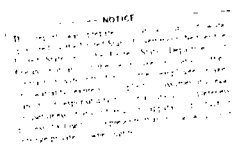
G. K. Patterson
R. B. Rozsa

September 5, 1978

Work performed under the auspices of the U S Department of Energy by the UCLLL under contract number W-7405-ENG-48

Distribution Category
UC 32

LAWRENCE LIVERMORE LABORATORY
University of California Livermore, California 94550



UCRL-52561

DYNSYL: A GENERAL-PURPOSE DYNAMIC SIMULATOR FOR CHEMICAL PROCESSES

G. K. Patterson*
R. B. Rozsa

MS. date: September 5, 1978

*Now with the Department of Chemical Engineering, University of Missouri-Rolla, Rolla, Missouri.

CONTENTS

DYNSYL: A GENERAL-PURPOSE DYNAMIC SIMULATOR FOR CHEMICAL PROCESSES

ABSTRACT

Lawrence Livermore Laboratory is conducting a safeguards program for the Nuclear Regulatory Commission. The goal of the Material Control Project of this program is to evaluate material control and accounting (MCA) methods in plants that handle special nuclear material (SNM). To this end we designed and implemented the dynamic chemical plant simulation program DYNSYL. This program can be used to generate process data or to provide estimates of process performance; it simulates both steady-state and dynamic behavior. The MCA methods that may have to be evaluated range from sophisticated on-line material trackers such as Kalman filter estimators, to relatively simple material balance procedures. This report describes the overall structure of DYNSYL and includes some example problems. The code is still in the experimental stage and revision is continuing.

INTRODUCTION

DYNSYL, a modification of DYNSYS,[1] uses modular program logic to simulate chemical plant dynamic behavior. The differential equations generated by each process unit module are timewise-integrated by a stiff equation system integrator. All equations can be integrated in a completely coupled mode, or the various units can be simulated in an uncoupled mode. Input data required include in and out process stream numbers, operating parameters (size, rate constants, operation mode, etc.), and stream parameters (flow rate, temperature, pressure, concentrations) for each unit, as well as graphical and printed output specifications, and simulation time specifications. Operator-initiated process changes may be input by terminal.

Output results include an input data echo, all stream parameter and unit parameter values at the end of each time interval, and printplot and plotter results for selected stream parameters as a function of time.

The program was developed to simulate chemical processes in the nuclear fuel cycle. The unit subroutines (modules) available are a general-purpose transport unit for equilibrium stage computations with heat transfer (liquid-liquid or liquid-vapor), or for stirred-tank mixing and reaction; a controller with various modes; a pipe; a pump; a highly accurate extractor for uranium and plutonium coextraction or separation in Purex plants; a plutonium precipitator; and a plutonium concentrator. Such simulations provide data for material accounting studies, particularly for on-line schemes, and allow studies of dynamic plant operation either for assessment or design.

PURPOSE

DYNSYL has been designed and implemented to help evaluate MCA methods in plants that handle SNM. To detect diversion of SNM from chemical processes using MCA methods, the process units and subsections of the plant must first be simulated in detail. Both the steady-state and dynamic behaviors of the chemical process are needed to accurately evaluate MCA methods.

The process simulation provides "measured data" for MCA methods evaluations. The MCA method is often simulated dynamically in much the same way that the process is simulated, with measured data as the input and probability of detection of a diversion as the output.

Figure 1 shows the steps of a MCA method evaluation. The process is first dynamically

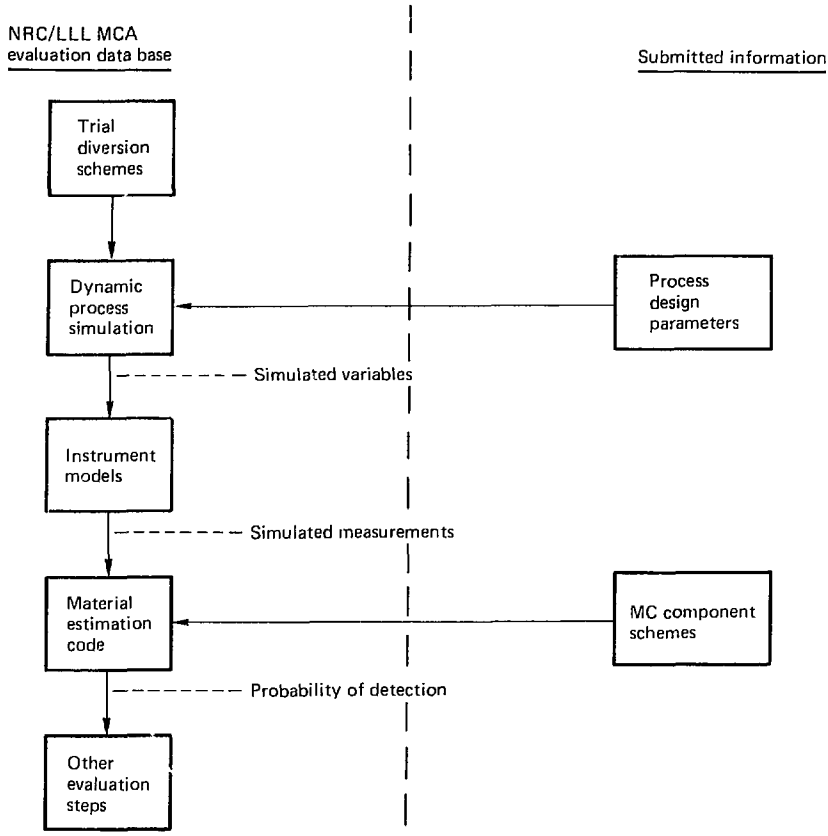


Fig. 1. Steps for evaluating an MCA method.

simulated using validated models of the process units and instruments that provide simulated measurements. These measured data are then input into the MCA method, and the probability of detection is derived for given levels of material diversion. The material diversions are incorporated into the process simulation as shown in Fig. 1.

As well as providing the basis for MCA evaluations, the dynamic and steady-state process simulations may also be used to check design validity and to obtain nominal operating levels for diversion tests.

RELATION TO PREVIOUS PROGRAMS

The dynamic chemical plant simulation we have described is a continuous timewise simulation, as opposed to discrete time interval types that simulate inventory and material flows in a plant (i.e., GPSS, [2] GASP, [3] Simscript [4]). Continuous dynamic simulations use differential and algebraic models of each of the units in a plant. These models are linked together to represent a numerical simulation of the analog behavior of the overall system.

Previous versions of continuous dynamic plant simulators include PRODYC, [5] DYSCO, [6] REMUS, [7] DYFLOW, [8] and DYNSYS, the model for DYNSYL. All of these simulation methods use generic models of process units that can be combined to simulate a larger section of a plant.

PRODYC interfaces the process unit simulations to the IBM Continuous System Modeling Program (CSMP [9]) to solve the system of differential equations generated. PRODYC, therefore, makes it easier to use CSMP dynamic process simulation. DYSCO is more modular in concept than PRODYC, allowing process flowsheet simulation entirely through input data specifications that consist primarily of a process topology matrix and unit and stream specifications. The code is formulated with process design as the major objective; graphical interaction and ease of process alteration are built-in.

DYNSYS was designed primarily as a batch simulation program with no interactive features included. It is otherwise basically the same as DYSCO with a topology matrix, specifications for each unit, and stream specifications that are also the main dynamic (state) variables in the simulation. DYNSYS does, however, account better for coupling effects between units as discussed in the next section.

PROGRAM MODULARITY

Two basic concepts are used to design dynamic chemical plant simulation codes: the equation-oriented and the modular methods. The equation-oriented method (sometimes called continuous system simulation) derives from analog simulation methods (CSMP, [9] MIMIC, [10] LEANS [11]) that provide solutions to large numbers of simultaneous interdependent differential equations. The method requires only that all the relevant equations be included. Generally, coupling between equations is included in each solution step.

The modular method uses computer code subroutines (modules) that compute the differentials of the state variables associated with the process unit simulated by the module. In most cases, equation solutions for each module are determined independently of the other modules during each time period. Thus, coupling between modules occurs only at the end of the time intervals and not at each step in the integration process as in most equation-oriented methods. As noted by Franks, [8] equation-oriented codes may be considered to represent a programming level intermediate between FORTRAN (or other scientific programming language) and modular codes that do not require mathematical formulations.

Both methods have advantages. Equation-oriented codes with complete equation coupling ensure the greatest dynamic simulation accuracy, particularly for large time steps, but require common storage for all derivative and/or variable values for all integration steps. This requirement can impose a severe limit on the number of plant units that may be simulated, particularly if the computer has small core storage. On the other hand, the modular codes with no intermodular coupling can be used with relatively small core storage as the commons in the code can be small, but accuracy may be poor unless the time intervals are very small. DYNSYS was written with partial intermodular coupling: the module equation solutions at each step of integration use coupling at all previous steps. DYNSYL, on the other hand, provides solutions with complete coupling at each present step. While developing DYNSYL, we endeavored to provide both completely coupled and uncoupled modes of solution to provide flexibility to the user in simulating large and small plants on his computer.

UNIT MODULE STRUCTURE

In DYNSYL, as in any modular chemical plant dynamic simulator, the unit module provides time-derivative values of the time-dependent variables (states) whenever they are needed by the integration section of the computer code. Because most unit simulations are lumped parameter simulations in which values of effluent stream variables equal the values of these variables in the unit (well-mixed approximation), only the stream variable time derivatives are computed. To have available all necessary state variable and parameter values, the unit modules must adequately communicate those values from the main program. In DYNSYL, as in DYNSYS, the commons rather than the subprogram arguments provide that communication. The commons contain the values of the state variables, the unit parameters, and the topology matrix, as well as other necessary values such as unit number. Figure 2 shows the makeup of the common statements in a DYNSYL module and a short description of the code.

After all variable and parameter values are provided in a module subprogram, the state variable derivative values may be computed. A certain amount of precomputation is usually needed, particularly if the module allows for several options. In addition, it is often beneficial to redefine variable names in terms of the usual names for the unit operation simulated, thus helping those unfamiliar with the program to understand its function.

Figure 3 shows the typical makeup of a DYNSYL module subprogram. Optional parts are shown in parentheses. As in DYNSYS, the module subprograms in DYNSYL are named as TYPE n where n is an integer. The subprograms are also identified in the executive part of the code with a descriptive name. That name should be given as a comment in the subprogram. Each subprogram should be prefaced with a complete description of its intended use, capabilities, limitations, and other information helpful to the user.

The values of the state variables are stored in the common array S(1, j, k). The array S(2, j, k) contains the corresponding values of the state variable derivatives. In the subprogram description presented in Fig. 1, the derivatives are first calculated as values of DERY (I). These values are then converted to new values in the array S(2, j, k).

```
C
      PROGRAM DYNSYL(INPUT,OUTPUT,TAPE2=INPUT,TAPE3=OUTPUT,
     *DATA,TAPE4=DATA)
      L=3000008
      CALL CREATE (4HDATA,L,ICO)
      CALL CHANGE(2H+A)
C
C     DYNSYL
C
C          DYNSYL IS AN LLL MODIFIED VERSION OF DYNSYS (BARNEY,
C     AHLUWALIA AND JOHNSON, FACULTY OF ENG. SCI., UNIV. OF WESTERN
C     ONTARIO, AUGUST 1975) WHICH UTILIZES MODULAR PROGRAM LOGIC TO
C     SIMULATE CHEMICAL PLANT DYNAMIC BEHAVIOR.  THE TIMEWISE INTEGRATION
C     OF THE DIFFERENTIAL EQUATIONS GENERATED BY EACH PROCESS UNIT MODULE
C     IS DONE BY THE LLL - PROGRAM DRIVE AND ITS SUB-PROGRAMS, A
C     STIFF SYSTEM INTEGRATOR WITH GREAT STABILITY AND ROBUSTNESS.  WITH
C     DRIVE AS THE EQUATION INTEGRATOR, ALL EQUATIONS GENERATED BY THE
C     UNIT MODULES ARE INTEGRATED TOGETHER.
C
C          PROGRAM NOTES
C
C          IT IS NECESSARY TO NUMBER THE STREAMS IN THE PLANT CONSECUTIVELY
C     WITH THE INPUT STREAMS FIRST ON THE LIST.  THIS ALLOWS THEM TO
C     BE EXCLUDED FROM THE INTEGRATION PROCESS.  THE NUMBERING OF UNITS
C     IN THE PLANT IS ARBITRARY, BUT THEY SHOULD BE 1 THRU NE - UNITS.  THE
C     NUMBER OF UNITS AND STREAMS WHICH MAY BE ACCOMODATED BY DYNSYL DEPENDS
C     ON THE DIMENSIONING OF COMMONS AND LOCAL RECORDS.  IT IS PRESENTLY
C     DIMENSIONED FOR 35 UNITS AND 45 STREAMS, BUT THAT COULD BE FEASIBLY
C     EXPANDED TO 45 UNITS AND 60 STREAMS TO RUN WITHIN A 500,000 WORD
C     IN-CORE LIMIT.  THE TIME REQUIRED FOR A RUN WITH 60 DYNAMIC VARIABLES
C     (10 STREAMS AND 6 VARIABLES) IS TYPICALLY 0.35 MINUTES.
C
C          PROGRAM DESCRIPTION
C
C          DYNSYL IS COMPOSED  OF MAIN.; INPUT PROGRAMS DYN1 AND GET; A
C     UNIT CALLING PROGRAM DYN2; OUTPUT PROGRAMS OUTPUTS, SAVEP, WRITEP,
```

Fig. 2. DYNSYL description and common blocks. (Continued on next page)

```
C     LPLOT, CALPLT(DD80 PLOT); COMMON PHYSICAL PROPERTY ROUTINES PROPS,
C     MOWE, CPLI, CPVA, ENTL, LAMB, DENL, DENV, VAPR, WILS, BUBL, KVAL, TEMPL,
C     AND TEMPV; INTEGRATION ROUTINES DRIVE, STIFF, PSET, INTERP, COSET, DEC,
C     AND SOL; A NOISE GENERATION PROGRAM NOISE; AND UNIT MODEL SUBROUTINES
C     STGIDL, CONTLR, PIPE, PUMP, EXTRTR, PRECIP, AND EVAPTR.  THE SUBROUTINE
C     STGIDL MAY BE USED AS AN IDEAL STAGE OF EXTRACTION OR ABSORPTION, A
C     FLASH DRUM WITH HEAT TRANSFER, A HEAT EXCHANGER OR A REACTOR.  EACH
C     UNIT SUBROUTINE MAY HAVE OTHER SUBROUTINES ASSOCIATED WITH IT WHICH
C     ARE NOT NAMED HERE.  EACH UNIT SUBROUTINE IS DESCRIBED IN MORE
C     DETAIL WITH THE SUBROUTINE ITSELF.  UNIT SUBROUTINES ARE EASILY ADDED, AS
C     DESCRIBED IN THE MANUAL.
C
C          THE COMMON BLOCK VARIABLES USED IN DYNSYL ARE:
C
C     COMMON/MAT/
C
C          MP(35,13) = UP TO 13 STREAM NUMBERS(IN +, OUT -) FOR EACH UNIT(UP
C                      TO 35)
C          EP(35,10) = UP TO 10 UNIT PARAMETER VALUES FOR EACH UNIT
C          S(2,45,11) = UP TO 11 STREAM VARIABLES FOR EACH OF 45 STREAMS
C                       WHEN THE FIRST INDEX IS 1; WHEN THE FIRST INDEX IS
C                       2, THE VALUES ARE TIME DERIVATIVES OF THE VARIABLES
C          EX(50) = EXTRA UNIT PARAMETERS (SEE MANUAL)
C
C     COMMON/COM/
C
C          NCOMP = MAXIMUM NUMBER OF COMPONENTS IN STREAMS
C          NC5 = NCOMP + 5
C          NC3 = NCOMP + 3
C          NE = NUMBER OF UNITS IN PLANT
C          NS = NUMBER OF STREAMS IN PLANT
C          TMAX = FINAL TIME OF SIMULATION
C          NB = NUMBER OF FEED STREAMS TO PLANT
C          N1 = FIRST COMPONENT WHICH IS DYNAMIC
C          N2 = LAST COMPONENT WHICH IS DYNAMIC
C
C     COMMON/PLT/
C
C          NPLOTS = NUMBER OF VARIABLES PLOTTED (LINES)
C          PLOTI = TIME INCREMENT BETWEEN SAVED POINTS
C          PLOTD(15,4) = PLOTTING SPECIFICATIONS FOR EACH OF 15 VARIABLES:
C                        STREAM NUMBER, STREAM VARIABLE, LOWER AND UPPER
C                        PLOT LIMITS
C          PLOTT = TIME VALUE OF LAST SAVED POINT
C          PTYPE = WIDTH OF PRINT PLOT: 0 FOR 100 SPACES, GT.0 FOR 50 SPACES
C          NSTATE = NUMBER OF DYNAMIC VARIABLES
C
C     COMMON/OUT/
C
C          NOUTPT = 1 FOR PRINTED OUTPUT AT EACH TIME INTERVAL
C          NLINE = 1 FOR A PRINT PLOT
C          NCAL = 1  FOR A DD80 PLOT(OR OTHER PLOT DEVICE)
C          NPR = NUMBER OF TIME INTERVALS PER SIMULATION
C          NSAVE = IF GT.0, POINTS WILL BE SAVED FOR PLOTTING
C          NNUMO = 1 FOR OUTPUT OF MEASURED DATA LIST(NSTATE VARIABLES)
C
C     COMMON/GERR/
C
C          JSTART = IF GT.0, PART OF SUBROUTINE SAVEP IS SKIPPED
C          TIME = CURRENT TIME VALUE
C          H = INITIAL INTEGRATOR INTERVAL
C          HINC = TIME INCREMENT VALUE
C          EPS = INTEGRATION PRECISION (ABSOLUTE)
C
C     COMMON/GRAPHC/
C
C          NPTS = NUMBER OF PLOTTED POINT
C          TPLOT(1500) = TIME VALUES(UP TO 1500)
C          YPLOT(1500,15) = UP TO 15 VARIABLE VALUES PLOTTED PER TIME VALUE
C          XPLOT(1000) = UP TO 1000 VARIABLE VALUES PRINTED AS MEASURED
C                        DATA PER TIME VALUE
C
```

Fig. 2. (Continued)

```
109 C      COMMON/IO/
110 C
111 C           NIN = INPUT DEVICE
112 C           NOUT = OUTPUT DEVICE FOR DATA ECHO AND NORMAL OUTPUT
113 C           NERR = OUTPUT DEVICE FOR ERROR MESSAGES AND LINE PLOT
114 C           NPOINT = OUTPUT DEVICE FOR MEASURED DATA
115 C
116 C      COMMON/UNIT/
117 C
118 C           IM = UNIT NUMBER
119 C           NMP = MAXIMUM NUMBEER OF STREAMS IN AND OUT OF ANY UNIT
120 C
121 C      COMMON/LARM/
122 C
123 C           STD = ABSOLUTE STANDARD DEVIATION OF PROCESS NOISE
124 C           PRCNT = PER CENT OF VARIABLE VALUE STANDARD DEVIATION OF
125 C                   PROCESS NOISE
126 C           EMC = SEED VALUE FOR RANDOM NUMBER GENERATOR
127 C           EMSTD = ABSOLUTE STANDARD DEVIATION OF MEASUREMENTS
128 C           EMPRCT = PER CENT OF VARIABLE VALUE STANDARD DEVIATION OF
129 C                    MEASUREMENTS
130 C
131        DIMENSION Y(300),YDOT(300)
132        COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
133        COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,N8,N1,N2,NF
134        COMMON /PLT/ NPLOTS,PLOTI,PLOTD(15,4),PLOTT,PTYPE,NSTATE
135        COMMON /OUT/ NOUTPT,NLINE,NCAL,NPR,NSAVE,NNUMO
136        COMMON /GERR/ JSTART,IMETH,TIME,H,HH,HINC,EPS,T,INTFL
137        COMMON /GRAPHC/ NPTS,TPLOT(1500),YPLOT(1500,15),XPLOT(1000)
138        COMMON /IO/ NIN,NOUT,NERR,NPOINT,NPRT
139        COMMON/CHANGE/IC,JC,CTIME
140        COMMON/UNIT/IM,NMP
141        COMMON/LARM/STD,PRCNT,EMC,EMSTD,EMPRCT
142        INTEGER OUTPRO,OUTSTO,OUTPRA,OUTSTA
143        DATA NIN,NOUT,NERR,NPOINT/2,3,3,4/
```

Fig. 2. (Continued)

```
      Subroutine Type n
c
c     (Subroutine descriptive name)
c
c     (description of subroutine, its capabilities, limitations, and other information helpful to users)
c
      COMMON/MAT/MP(35,13),EP(35,10),S(2,45,13),EX(50)

      COMMON/CON/JCOMP,NCS,NE,NS,TMAX,NC3,NB,N1ST,NFIN

      COMMON/GERR/JSTART,IMETH,TIME,H,HINC,EPS,TT

      COMMON/UNIT/IM,NMP

      (REAL variable list)

      (INTEGER variable list)

      (DIMENSION variable list)
c
c     (equipment parameter list, component list, etc.)

      IF (INTFL.EQ.0)* GO TO 2

      RETURN

      2 CONTINUE

      (Rename variables from S-array†, equipment parameters from EP-array, stream names from MP-array.)

      (Do precomputations necessary to compute derivatives of state variables at present time.)

      DO nn I=1, number of state variables                          ]
                                                                    | Used when this module
      DERY (I) = function of state variables, equipment parameters  | called by DRIVE with DYN2.
                                                                    ]
nn    CONTINUE

      OTIME=TIME-HINC                                               ]
                                                                    | Used when this module called
      INDEX=1                                                       | by MAIN with DYN2. Another
                                                                    | subroutine must be called by
      CALL DRIVELIN,OTIME,H,Y,TIME,ESP,IMETH,INDEX)**               | DRIVEL to compute derivatives
                                                                    ] of Y, usually named DERY.
      (post-computations necessary)

      (rename derivative values with S-array names)

      RETURN

      END
```

*If the subroutine calls the integrator this statement would be IF(INTFL.EQ.1)GO TO 2.
†*S(1,j,k) are state variable values; S(2,j,k) are their derivatives; S(1,j,k) maps to Y(i) for the integrator; S(2,j,k) maps to DERY(i).*
**The arguments are:
N – number of state variables
OTIME – beginning time of interval
H – integration step size
Y – state variable name
TIME – final time of interval
EPS – convergence criterion (fraction of variable)
IMETH – integration method (see integrator manual[14])

Fig. 3. Unit module subprogram description.

THE INTEGRATOR

The integrator used in DYNSYS is a modification of the original GEAR [12,13] integration code for stiff systems of differential equations. It is a subroutine called by each unit module subprogram to integrate the differential equations represented by the differentials of the state variables. In DYNSYS the integrator is a two-step predictor-corrector with a maximum order of six. The integration is done either by an Adam-Bashforth method requiring storage of previous derivative values, or by a "stiff" method requiring storage of previous state variable values. The integrator stores the values of all differential or state variables up to the order six from all the unit module subprograms. For the present time interval, all state variable values determined by the predictor step are available for use in the corrector step. Therefore, the previous state variable values account for coupling between equations, which represents all units in the simulated plant.

Besides including intermodular coupling, the GEAR-based integrator in DYNSYS is designed to handle stiff sets of equations, those with a wide range of time constants in the various differential equations integrated. Unfortunately, use of DYNSYS with the unit module STGIDL (see Appendix A) caused many unstable cases that were impossible to integrate. For that reason an improved integration subprogram was sought for DYNSYL.

The subprogram chosen for DYNSYL is a version of the GEAR stiff integrator written for use at LLL. [14] It was designed to serve as the integrator in locally generated, continuous-system simulations. Its main program, DRIVE, is called by the simulator. DRIVE then calls STIFF which calls a user-written subprogram to generate values of the differentials to be integrated. When used with

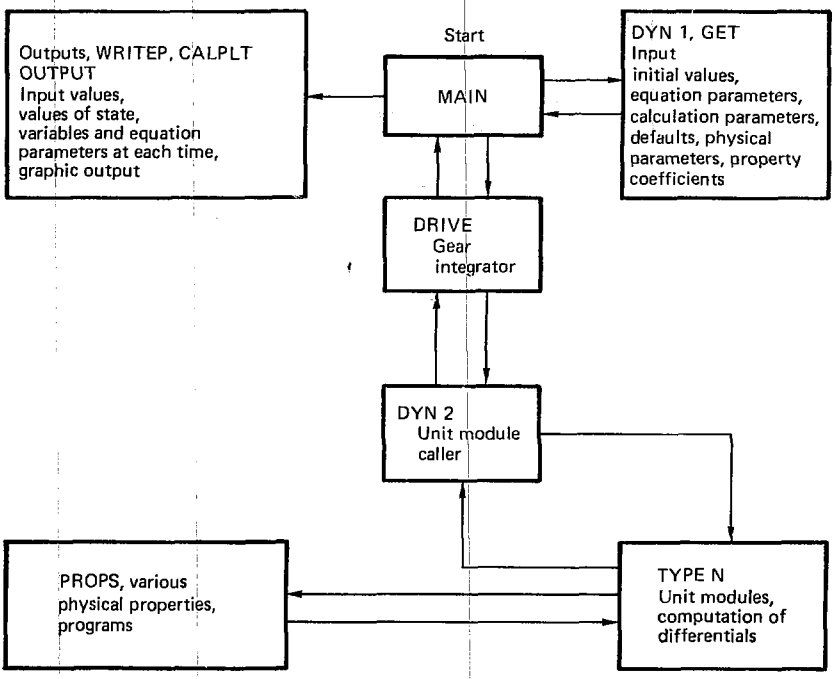


Fig. 4. Coupled mode of operation of DYNSYL (general structure).

DYNSYL, the LLL version of GEAR is called by the main program of DYNSYL. GEAR then calls for values of each of the differentials in the unit modules through an ordering program called DYN2 (see Fig. 4). Under these circumstances the integrator uses present values of all differentials at each integration step.

An alternate use of GEAR in DYNSYL is designed to use less memory capacity. In that mode the integrator is called by each individual unit module subprogram. The commons are then only as large as the largest subprogram common because all state variable values for all orders are not stored. The alternate mode accounts for intermodular coupling only through the state variable values from the last time step.

To efficiently solve large sets of differential equations, the GEAR integrator can use a Newton-Raphson method requiring Jacobian values for the state variables. The integrator manual [14] explains all the options for its use, but for the purposes of DYNSYL we only considered two: analytical computation of the Jacobian values in the unit module subprograms, and numerical computation of the Jacobian values by the integrator. The former choice saves computer time but the latter is more flexible, as analytical Jacobian expressions can, in many cases, be almost impossible to derive. So far, we have only used the numerical option.

OVERALL PROGRAM STRUCTURE

The general structure of DYNSYL is shown in Fig. 4. The main program first calls a data-reading program, DYN1, which also uses GET, which, in turn, primarily reads alphameric information. The main program calls the integrator program DRIVE (after all necessary precomputations have been made); DRIVE calls DYN2, the unit module program caller, through STIFF, a subprogram of DRIVE. To determine updated values of the state variables after each time increment, the unit module programs compute values of the derivatives of the state variables to be used by the integrator program. The unit module programs call the physical property programs when necessary. At the end of each output interval (several integrator time intervals), the integrator returns to the main program so that the output programs may be called.

The structure shown in Fig. 4 is for complete coupling of all equations of all units. If it is desired to save memory space in the computer, or if a very large chemical plant is to be simulated, an uncoupled mode may be used. The structure of that mode is shown in Fig. 5.

The uncoupled mode is not nearly as efficient as the coupled mode, nor is it as accurate, because the integrator must be initialized at the beginning of each time step of MAIN if the integrator is being called by more than one unit module. The multiple-order capabilities of the integrator are also interrupted at each time step of MAIN.

The logic used in either the coupled or uncoupled mode is shown in Fig. 6, which shows the detailed organization of DYNSYL primarily through subroutine calls.

PROCESS TOPOLOGY AND PROCESS VARIABLES

To transmit to the unit module subprograms those process streams connected to them, it is necessary to define a process topology array, MP (i,j), whose values are the numbers of the streams defined within the unit modules. A unit module might, for instance, have three streams connected to it numbered 1, 12, and 33. If the unit in the plant is number 2, then the topology matrix values would be MP(2,1) = 1, MP(2,2) = 12, and MP(2,3) = 33. Such an array makes possible greater flexibility than if the stream numbers are assigned within the unit modules during each use of the simulation code.

The values of the process variables are contained in the array S(i,j,k). As mentioned previously, when i = 1, the values are the state variable values; when i = 2, the values are the derivatives computed in the unit modules. The value of j is the stream number. The values of k indicate the particular stream variables as follows: 3 for flow rate, 4 for temperature, 5 for pressure, and 6 to n for concentrations of various components, where n is the maximum dimension of K. The value of the stream number is S(1,j,1) and S(1,j,2) is a flag whose value may be positive or negative. If the stream flag

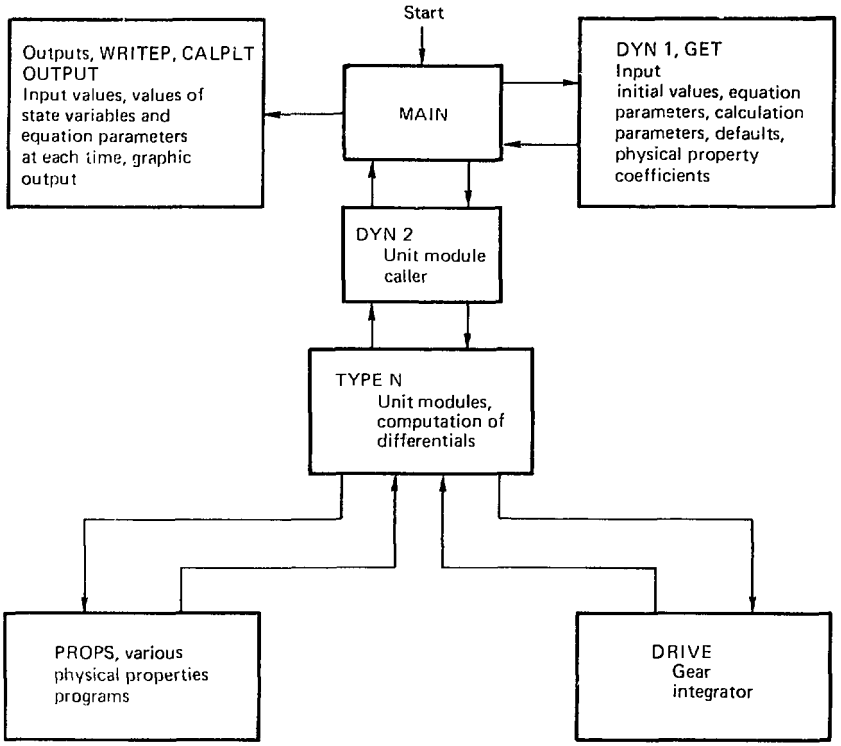


Fig. 5. Uncoupled mode of operation of DYNSYL (general structure).

is negative, that stream vector is suppressed in the output.

Equipment and other parameter values are contained in the array EP(i,j), where i is unit number and j is parameter number. Each unit may have up to 10 parameters in that array, plus additional ones in vector EX(i). That vector will be discussed further in the section on INPUT.

VARIABLES AND ARRAYS

The process topology, variable, and equipment parameter arrays are the main common arrays used in DYNSYL. Many other arrays and common variables are defined in the listing from the main DYNSYL program shown in Fig. 6. Most of these are self-explanatory or become clear with use of the DYNSYL code, and thus we will not describe them here.

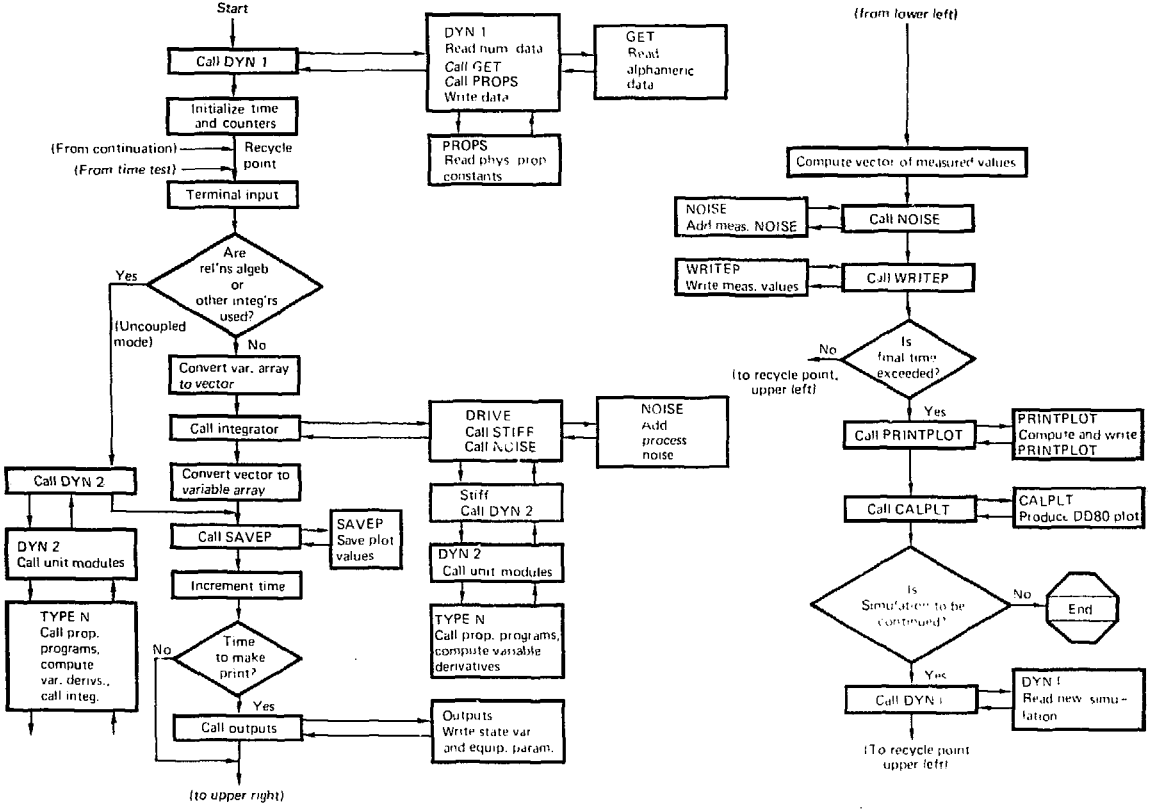


Fig. 6. Organization of DYNSYL.

ERROR MESSAGES

The error messages generated by DYNSYL are a combination of those from the original DYNSYS and from the GEAR-integrator code. The messages are usually clear, but the proper remedy may not be. The DYNSYS manual [1] and the integrator manual [14] contain suggested responses to each of the error messages that may be generated by the code.

PROGRAM EXECUTION

Upon execution of the DYNSYL code, the coded input values are first read by subroutines DYN1 and GE1. The user is then immediately informed of the TIME value and asked for the TIME value for the first change-in-state variables. Such changes may be made to simulate process upsets, material diversions, or control actions. The requested TIME value is input through the terminal, and execution resumes.

The user is informed when it is time for a change, and is asked to type the number of state variables to be changed. He is then asked for the stream and variable numbers as well as the new variable value for each change. The proper format for the input is always given. After all changes are given, the user is asked for the next TIME value for which a change will be made. Whenever that TIME value exceeds the simulation time, execution is completed.

The files containing numerical and graphical output are created when the execution is completed. The output devices from which these files may be obtained by the user depend on computer type and user specifications.

INPUT

The form of the input specifications for DYNSYL is nearly the same as in DYNSYS. (For a very complete description of the input system used, the DYNSYS manual [1] may also be consulted.) To illustrate the input format and some of the variations, the example input shown in Fig. 7 will be discussed below. The example is for simulation of a 14-stage extractor, which is shown schematically in Fig. 8.

One may analyze the necessary format for the input data by reviewing DYN1, given in the listing in Appendix B. There are several default values that are first established. These are listed in Table 1.

To override the default values, input values may be read as shown in Table 2 [FORMAT (3A4, F12.5)].

A new unit module not already included as part of the module library can be inserted into the code through the input data. The alphameric LIBRARY signals new module-name input as shown in the example in Fig. 8. The number after LIBRARY is the number of new modules. The next lines are the new module names and their numbers. Note that all numerical input is floating-point even when the numbers are used as integers. The formats are all (3A4, F12.5).

The alphameric PROCESS signals the beginning of equipment and process data. An input listing is required, consisting of a unit module name and the unit number followed by the stream numbers (in and out of the unit) and equipment parameters in the proper order (specified by the unit module subprogram). If unit numbers are negative, equipment parameters will be output. If stream numbers are negative, stream flows are outward. The formats are (3A4, F12.5) for unit name and number and (12X, 5F12.5) for the stream numbers and equipment parameters.

After the alphameric END, the number of streams and stream variable values at zero time are input. The stream variable values should be given in the order indicated for the state variable array discussed previously, with the stream number and flag as the first two. If the stream flag is negative, output is suppressed. Again the format is (12X, 5F12.5). The alphameric EXPLICIT or SPECIAL comes before each stream variable list. EXPLICIT indicates that values for a particular stream are listed. SPECIAL followed by a stream number indicates that values for that stream number through the stream number indicated in the array list are given. SPECIAL allows input of common values for a number of streams using one input list.

After the alphameric END, PROPERTIES indicates that coefficients for several different

property correlation equations for each component will be read. The number of property equations depends on the number following PROPERTIES. If -1 or -2 follows PROPERTIES, property coefficients for water or air, respectively, are assumed by the program. The property equation coefficient input in DYNSYL is the same as in DYNSYS; see Ref. 1 for details. The formats are (3A4, F12.5) and (12X, 5F12.5).

After another alphameric END, GRAPH indicates with a subsequent number several graph lines (variables) to be plotted by CALPLT or LPLOT. The two numbers on the next line indicate the abscissa increment and the lineplot width. If the second number is 1.0, the lineplot is 50 spaces wide; if it is 0.0, the lineplot is 100 spaces wide. The next lines give the specifications for each variable graphed, the stream number and variable, and the lower and upper ordinate limits.

If the user desires to continue with a different number of iterations in a given time, or with a different number of iterations between printing, he follows the procedure that we showed in the example. The time given is the final time, and the number of iterations is the number as if beginning at zero time. The prompt END must be given after each continuation specification and also to end the simulation.

```
*****                    **************                    *****
TEST SIMULATION OF A 14-STAGE EXTRACTION COLUMN USING STGIDL
*****                    **************                    *****
BEGIN
COMPONENTS        4.0
IN/OUT             10.0
TIME              100.0
NO OF ITRTNS      500.0
FEED STRMS          1.0
COUPLED STMS       33.0
FIRST COMP          4.0
LAST COMP           7.0
ITER BTW PRT       10.0
NOISE STD DV     1.0E-3
PERCNT NOISE        0.0
MEAS NOISE S     1.0E-2
M NOISE PRCT        0.0
OUTPUT
NUMOUTPUT
CALPLOT
LIBRARY             1.0
TUBRXR             16.
PROCESS
STGIDL             -1.0
                   15.0          0.0           0.0         -16.0          17.0
                    0.0          0.0         -18.0
                    0.0         13.7          13.7         100.0         100.0
                   0.30
STGIDL             -2.0
                   14.0          0.0           0.0         -15.0          18.0
                    0.0          0.0         -19.0
                    0.0         13.7          13.7         100.0         100.0
                   0.30
STGIDL             -3.0
                   13.0          0.0           0.0         -14.0          19.0
                    0.0          0.0         -20.0
                    0.0         13.7          13.7         100.0         100.0
                   0.30
STGIDL             -4.0
                   12.0          0.0           0.0         -13.0          20.0
                    0.0          0.0         -21.0
                    0.0         13.7          13.7         100.0         100.0
                   0.30
STGIDL             -5.0
                   11.0          0.0           0.0         -12.0          21.0
                    0.0          0.0         -22.0
                    0.0         13.7          13.7         100.0         100.0
                   0.30
STGIDL             -6.0
                   10.0          0.0           0.0         -11.0          22.0
                    0.0          0.0         -23.0
                    0.0         13.7          13.7         100.0         100.0
                   0.30
STGIDL              7.0
                    9.0          0.0           0.0         -10.0          23.0
                    1.0          0.0         -24.0
                    0.0         13.7          13.7         100.0         100.0
                   0.30
STGIDL             -8.0
                    8.0          0.0           0.0          -9.0          24.0
```

Fig. 7. Example input for a 14-stage extractor with diversion on input line. (Continued on next page)

```
                    0.0          0.0         -25.0
                    0.0         13.7          13.7        100.0       100.0
                   0.30
STGIDL             -9.0
                    7.0          0.0           0.0         -8.0        25.0
                    0.0          0.0         -26.0
                    0.0         13.7          13.7        100.0       100.0
                   0.30
STGIOL            -10.0
                    6.0          0.0           0.0         -7.0        26.0
                    0.0          0.0         -27.0
                    0.0         13.7          13.7        100.0       100.0
                   0.30
STGIOL            -11.0
                    5.0          0.0           0.0         -6.0        27.0
                    0.0          0.0         -28.0
                    0.0         13.7          13.7        100.0       100.0
                   0.30
STGIOL            -12.0
                    4.0          0.0           0.0         -5.0        28.0
                    0.0          0.0         -29.0
                    0.0         13.7          13.7        100.0       100.0
                   0.30
STGIDL            -13.0
                    3.0          0.0           0.0         -4.0        29.0
                    0.0          0.0         -30.0
                    0.0         13.7          13.7        100.0       100.0
                   0.30
STGIDL            -14.0
                    2.0          0.0           0.0         -3.0        30.0
                    0.0          0.0         -31.0
                    0.0         13.7          13.7        100.0       100.0
                   0.30
STGIDL            -15.0
                   32.0          0.0          -1.0        -33.0         0.0
                    0.0          0.0           0.0          0.0         0.0
                    0.0          0.1           0.0          0.0         0.0
                    0.0
END
STREAMS            33.0
EXPLICIT
                    1.0          1.0          7.28         25.0         1.0
                    9.5         4.78           3.1          0.0
SPECIAL             2.0
                   16.0          1.0          2.43         25.0         1.0

SPECIAL            17.0
                   23.0          1.0          1.02         25.0         1.0

SPECIAL            24.0
                   31.0          1.0           8.3         25.0         1.0

EXPLICIT
                   32.0          1.0          7.28         25.0         1.0
                    9.5         4.78           3.1          0.0
EXPLICIT
                   33.0          1.0           0.0         25.0         1.0

END
PROPERTIES         -1.0


END
GRAPH               4.0
                    2.0          0.0
                   16.0          7.0           0.0         20.0
                   31.0          7.0           0.0          1.0
                    1.0          3.0           0.0         10.0
                    1.0          7.0           0.0         10.0
CONTINUE
TIME            1000.0
NO OF ITRTNS     100.0
ITER BTW PRT       2.00
END
CONTINUE
TIME           10000.0
NO OF ITRTNS     500.0
ITER BTW PRT      10.0
END
END
```

Fig. 7. (continued)

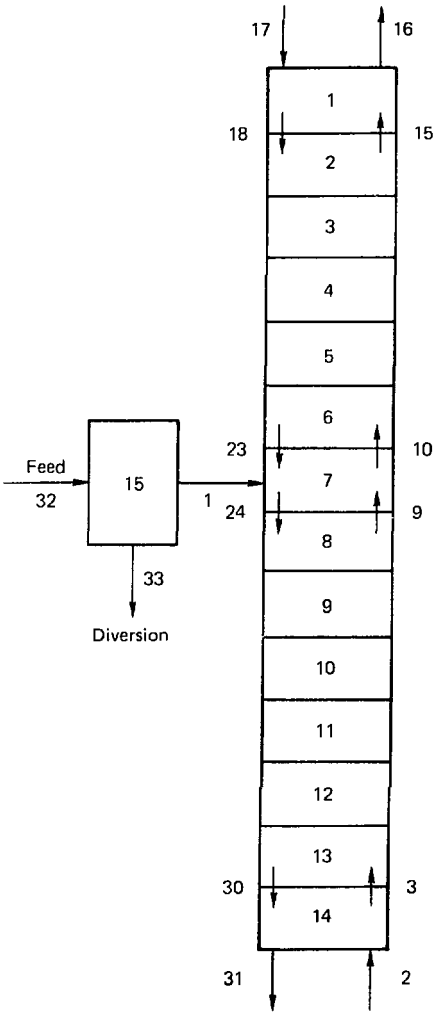


Fig. 8. Diagram of example extraction column. Unit 15 is of nearly zero volume for diversion simulation.

Table 1. Input default values.

Input variables	Definition
EMC=0.0	Argument for random number generator
N1ST=1	First variable to be a state variable
NFIN=6	Last variable to be a state variable
NB=1	Number of nonvariable streams
MAXNE=35	Maximum number of unit
TIME=0.0	Initial time value
ISTIFF=2	Stiff method of integration; 1 for Adams Method
MITER=2	Numerically computed Jacobian; see DRIVE in listing
NCOMP=1	Number of components
NMP=5	Maximum number of streams in and out of a unit
H=E-8	Initial time increment
TMAX=10.0	Simulation time
NPR=1	Number of iterations to be run
EPS=0.001	Convergence criterion
NOUTPT=0	No printout of state variable values
NNUMO=0	No printout of measurement vector values
NLINE=0	No lineplot
NCAL=0	No DD80 plot
NOMES=0	No nonfatal error messages
STD=0.0	Standard deviation of process noise
PRCNT=0.0	STD as percent of state variable value
EMSTD=0.0	Standard deviation of measurement noise
EMPRCT=0.0	EMSTD as percent of measured variable value
NRPT=1	Number of iterations between printing.

Table 2. Input values.

Input variable	Numerical value	Definition
BEGIN		*Starts input process; all preceeding labels are ignored*
NONSTIFF		Causes nonstiff integration
COMPONENTS		Number of components
IN OUT		Number of streams in and out of unit
FEED STREAMS		Number of nonvariable streams
FIRST VAR	X	First variable to be a state variable
LAST VAR	X	Last variable to be a state variable
DELT	X	Initial time increment
TIME	X	Simulation time
NO OF ITER	X	Number of interations to be run
TOLERANCE	X	Convergence criterion
ITER BTW PRT	X	*Number of iterations between printing*
NOISE	X	Standard deviation of process noise
PERCENT NOIS	X	Above as percent of state variable value
MEAS NOISE	X	Standard deviation of measurement noise
M NOISE PRCT	X	Above as percent of state variable value
OUTPUT	X	Print state variable values
NUMOUTPUT	X	Print measurement vector values
LINEPLOT	X	Print plot to be output
CALPLOT		DD80 plot to be produced
NOMESSAGE		Print nonfatal error messages
INTEG METHOD	X	Value of MITER in integrator.[a]

[a]MITER=2 causes Jacobian values to be numerically determined.

OUTPUT

The output of DYNSYL begins with an echo of the input in almost the same form, as shown in Fig. 9. At each print time beginning with TIME=0.0, all stream variables except for suppressed streams (negative stream number in STREAMS input) are printed. Figure 9 shows a sampling of output at various times. For each negative unit number the equipment specifications are also given at each print time. Any changes in stream variables or equipment parameters by interactive input are, of course, reflected in the output at the next print time.

If a lineplot is specified, it will follow the state variable and equipment parameter output after each simulation time (initial time and each continuation). Some care must be exercised in specifying the time (abscissa) spacing, as the lineplot can become excessively long.

The DD80 plot produced by CALPLT is placed on disk under the name PLTX. A separate plot is produced for each continuation of the simulation. That form of output is frequently the most useful for comparisons of various run conditions. Examples for the input and output are shown in Figs. 10 through 12.

Another form of output produced by DYNSYL is tailored for online data analysis methods such as Kalman filters. The measurement data vector is computed in the MAIN program. The vector may contain any consecutive list of variables from each stream that is not an input (constant) stream. Values from each interaction of MAIN are printed by subroutine WRITEP when NUMOUTPUT is specified in the input. The form of the output is simply the time value followed by each state variable value in order with a format of (6F12.3).

```
*****                    **************                          ******
TEST SIMULATION OF A 14-STAGE EXTRACTION COLUMN USING STGIDL
*****                    **************                          *****
BEGIN
COMPONENTS          4.00000
IN/OUT             10.00000
TIME              100.00000
NO OF ITRTNS      500.00000
FEED STRMS          1.00000
COUPLED STMS       33.00000
FIRST COMP          4.00000
LAST COMP           7.00000
ITER BTW PRT       10.00000
NOISE STD DV        0.00100
PERCNT NOISE
MEAS NOISE S
M NOISE PRCT        0.01000
OUTPUT
NUMOUTPUT
CALPLOT
PROCESS
SIGIDL                 -1
UNIT -1   TYPE   1
                15.00000          0.                0.          -16.00000          17.00000
                    0.            0.          -18.00000            -0.              -0.
                    0.          13.70000        13.70000         100.00000        100.00000
                 0.30000          -0.               -0.              -0.              -0.
STGIDL                 -2
UNIT -2   TYPE   1
                14.00000          0.                0.          -15.00000          18.00000
                    0.            0.          -19.00000            -0.              -0.
                    0.          13.70000        13.70000         100.00000        100.00000
                 0.30000          -0.               -0.              -0.              -0.
STGIDL                 -3
UNIT -3   TYPE   1
                13.00000          0.                0.          -14.00000          19.00000
                    0.            0.          -20.00000            -0.              -0.
                    0.          13.70000        13.70000         100.00000        100.00000
                 0.30000          -0.               -0.              -0.              -0.
STGIDL                 -4
UNIT -4   TYPE   1
                12.00000          0.                0.          -13.00000          20.00000
                    0.            0.          -21.00000            -0.              -0.
                    0.          13.70000        13.70000         100.00000        100.00000
                 0.30000          -0.               -0.              -0.              -0.
STGIDL                 -5
UNIT -5   TYPE   1
                11.00000          0.                0.          -12.00000          21.00000
                    0.            0.          -22.00000            -0.              -0.
                    0.          13.70000        13.70000         100.00000        100.00000
                 0.30000          -0.               -0.              -0.              -0.
STGIDL                 -6
UNIT -6   TYPE   1
                10.00000          0.                0.          -11.00000          22.00000
                    0.            0.          -23.00000            -0.              -0.
                    0.          13.70000        13.70000         100.00000        100.00000
                 0.30000          -0.               -0.              -0.              -0.
STGIDL                  7
UNIT  7   TYPE   1
```

Fig. 9. Example output for a 14-stage extractor with divertor on input line. (Continued on next page)

```
                  9.00000          0.              0.             -10.00000       23.00000
                  1.00000          0               0               -0.             -0.
                  0.              13.70000        13.70000        100.00000       100.00000
                  0.30000          -0              -0.             -0.             -0.
STGIDL
UNIT -8    TYPE   1
                  8.00000          0.              0.              -9.00000        24.00000
                  0.               0.            -25.00000         -0.             -0.
                  0.              13.70000        13.70000        100.00000       100.00000
                  0.30000          -0              -0.             -0.             -0.
STGIDL
UNIT -9    TYPE   1
                  7.00000          0.              0.              -8.00000        25.00000
                  0.               0               0.              -0.             -0.
                  0.              13.70000        13.70000        100.00000       100.00000
                  0.30000          -0.             -0.             -0.             -0.
STGIDL
UNIT-10    TYPE   1
                  6.00000          0.              0.              -7.00000        26.00000
                  0.               0.            -27.00000         -0.             -0.
                  0.              13.70000        13.70000        100.00000       100.00000
                  0.30000          -0.             -0.             -0.             -0.
STGIDL
UNIT-11    TYPE   1
                  5.00000          0.              0.              -6.00000        27.00000
                  0.               0.            -28.00000         -0.             -0.
                  0.              13.70000        13.70000        100.00000       100.00000
                  0.30000          -0.             -0.             -0.             -0.
STGIDL
UNIT-12    TYPE   1
                  4.00000          0.              0.              -5.00000        28.00000
                  0.               0               0.              -0.             -0.
                  0.              13.70000        13.70000        100.00000       100.00000
                  0.30000          -0              -0.             -0.             -0.
STGIDL
UNIT-13    TYPE   1
                  3.00000          0               0.              -4.00000        29.00000
                  0.               0.            -30.00000         -0.             -0.
                  0.              13.70000        13.70000        100.00000       100.00000
                  0.30000          -0.             -0.             -0.             -0.
STGIDL
UNIT-14    TYPE   1
                  2.00000          0.              0.              -3.00000        30.00000
                  0                0               0.              -0.             -0.
                  0               13.70000        13.70000        100.00000       100.00000
                  0.30000          -0.             -0.             -0.             -0.
STGIDL
UNIT-15    TYPE   1
                 32.00000          0.             -1.00000        -33.00000        0.
                  0                0               0.              0.              0.
                  0.               0.10000         0.              0.              0.
                  0                -0              -0.             -0.             -0.

END
STREAMS                33
EXPLICIT
SPECIAL
SPECIAL
SPECIAL
EXPLICIT
EXPLICIT

END
STREAM            1.00000          1.00000         7.28000         25.00000        1.00000
                  9.50000          4.76000         3.10000          0.
STREAM            2.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM            3.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM            4.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM            5.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM            6.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM            7.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM            8.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM            9.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM           10.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM           11.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM           12.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM           13.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM           14.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM           15.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
STREAM           16.00000          1.00000         2.43000         25.00000        1.00000
                  -0.              -0.             -0.             -0.
```

Fig. 9. (Continued)

```
STREAM          17.00000     1.00000     1.02000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          18.00000     1.00000     1.02000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          19.00000     1.00000     1.02000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          20.00000     1.00000     1.02000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          21.00000     1.00000     1.02000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          22.00000     1.00000     1.02000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          23.00000     1.00000     1.02000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          24.00000     1.00000     8.30000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          25.00000     1.00000     8.30000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          26.00000     1.00000     8.30000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          27.00000     1.00000     8.30000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          28.00000     1.00000     8.30000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          29.00000     1.00000     8.30000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          30.00000     1.00000     8.30000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          31.00000     1.00000     8.30000    25.00000     1.00000
                   -0.          -0.         -0.        -0.
STREAM          32.00000     1.00000     7.28000    25.00000     1.00000
                 9.50000     4.78000     3.10000        0.
STREAM          33.00000     1.00000        0.      25.00000     1.00000
                   -0.          -0.         -0.        -0.
PROPERTIES      -1.00000
PROPERTIES AS WATER
END
GRAPH                4
                 2.00000          0.
                16.00000     7.00000          0.    20.00000
                31.00000     7.00000          0.     1.00000
                 1.00000     3.00000          0.    10.00000
                 1.00000     7.00000          0.    10.00000
```

```
$$$$$     PROCESS VARIABLES AT TIME =    0.              $$$$$
STREAM      FLOW      TEMP      PRES    COMPONENTS
     1      7.280     25.0       1.0     9.50000    4.78000    3.10000    0.
     2      2.430     25.0       1.0     -0.        -0.        -0.        -0.
     3      2.430     25.0       1.0     -0.        -0.        -0.        -0.
     4      2.430     25.0       1.0     -0.        -0.        -0.        -0.
     5      2.430     25.0       1.0     -0.        -0.        -0.        -0.
     6      2.430     25.0       1.0     -0.        -0.        -0.        -0.
     7      2.430     25.0       1.0     -0.        -0.        -0.        -0.
     8      2.430     25.0       1.0     -0.        -0.        -0.        -0.
     9      2.430     25.0       1.0     -0.        -0.        -0.        -0.
    10      2.430     25.0       1.0     -0.        -0.        -0.        -0.
    11      2.430     25.0       1.0     -0.        -0.        -0.        -0.
    12      2.430     25.0       1.0     -0.        -0.        -0.        -0.
    13      2.430     25.0       1.0     -0.        -0.        -0.        -0.
    14      2.430     25.0       1.0     -0.        -0.        -0.        -0.
    15      2.430     25.0       1.0     -0.        -0.        -0.        -0.
    16      2.430     25.0       1.0     -0.        -0.        -0.        -0.
    17      1.020     25.0       1.0     -0.        -0.        -0.        -0.
    18      1.020     25.0       1.0     -0.        -0.        -0.        -0.
    19      1.020     25.0       1.0     -0.        -0.        -0.        -0.
    20      1.020     25.0       1.0     -0.        -0.        -0.        -0.
    21      1.020     25.0       1.0     -0.        -0.        -0.        -0.
    22      1.020     25.0       1.0     -0.        -0.        -0.        -0.
    23      1.020     25.0       1.0     -0.        -0.        -0.        -0.
    24      8.300     25.0       1.0     -0.        -0.        -0.        -0.
    25      8.300     25.0       1.0     -0.        -0.        -0.        -0.
    26      8.300     25.0       1.0     -0.        -0.        -0.        -0.
    27      8.300     25.0       1.0     -0.        -0.        -0.        -0.
    28      8.300     25.0       1.0     -0.        -0.        -0.        -0.
    29      8.300     25.0       1.0     -0.        -0.        -0.        -0.
    30      8.300     25.0       1.0     -0.        -0.        -0.        -0.
    31      8.300     25.0       1.0     -0.        -0.        -0.        -0.
    32      7.280     25.0       1.0     9.50000    4.78000    3.10000    0.
    33      0.        25.0       1.0     -0.        -0.        -0.        -0.

0       1           0.          13.70000        13.70000        100.00000       100.00000
0       2           0.          13.70000        13.70000        100.00000       100.00000
0       3           0.          13.70000        13.70000        100.00000       100.00000
0       4           0.          13.70000        13.70000        100.00000       100.00000
0       5           0.          13.70000        13.70000        100.00000       100.00000
0       6           0.          13.70000        13.70000        100.00000       100.00000
0       8           0.          13.70000        13.70000        100.00000       100.00000
```

Fig. 9. (Continued)

0	9	0.	13.70000	13.70000	100.00000	100.00000
0	10	0.	13.70000	13.70000	100.00000	100.00000
0	11	0	13.70000	13.70000	100.00000	100.00000
0	12	0	13.70000	13.70000	100.00000	100.00000
0	13	0	13.70000	13.70000	100.00000	100.00000
0	14	0	13.70000	13.70000	100.00000	100.00000
0	15	.	0.10000	0.	0.	0.

$$$$$ STREAM	FLOW	TEMP	PRES	PROCESS VARIABLES AT TIME = 2.00000E+00 COMPONENTS			$$$$$
1	7.280	25.0	1.0	9.50000	4.78000	3.10000	0.
2	2.430	25.0	1.0	0.00107	-0.00111	-0.00046	-0.00119
3	2.430	25.0	1.0	0.00267	-0.00048	-0.00072	0.00054
4	2.430	25.0	1.0	0.00086	-0.00051	0.00011	-0.00042
5	2.430	25.0	1.0	0.00061	-0.00194	0.00176	-0.00013
6	2.430	25.0	1.0	0.00061	0.00090	-0.00010	-0.00009
7	2.430	25.0	1.0	0.00359	0.00111	0.00142	-0.00067
8	2.430	25.0	1.0	0.03442	0.02695	0.01734	-0.00112
9	2.430	25.0	1.0	0.43501	0.30044	0.11532	-0.00120
10	2.430	25.0	1.0	7.49684	3.06977	0.33041	-0.00015
11	2.430	25.0	1.0	0.24063	0.07634	0.01009	0.00224
12	2.430	25.0	1.0	0.00548	0.00121	0.00072	0.00057
13	2.430	25.0	1.0	0.00071	0.00100	0.00006	-0.00203
14	2.430	25.0	1.0	-0.00125	0.00012	-0.00063	0.00173
15	2.430	25.0	1.0	-0.00130	-0.00115	-0.00125	0.00039
16	2.430	25.0	1.0	-0.00032	0.00044	0.00120	-0.00023
17	1.020	25.0	1.0	-0.00021	-0.00036	-0.00088	0.00089
18	1.020	25.0	1.0	0.00089	-0.00216	0.00086	0.00002
19	1.020	25.0	1.0	0.00001	0.00138	-0.00052	0.00118
20	1.020	25.0	1.0	-0.00049	0.00142	-0.00106	-0.00091
21	1.020	25.0	1.0	0.00068	0.00050	-0.00123	0.00187
22	1.020	25.0	1.0	0.02311	0.00745	0.00052	-0.00033
23	1.020	25.0	1.0	0.98835	0.38975	0.04683	0.00036
24	8.300	25.0	1.0	0.64955	0.77283	1.69446	-0.00043
25	8.300	25.0	1.0	0.11272	0.24023	0.73128	0.00010
26	8.300	25.0	1.0	0.05294	0.11694	0.25360	0.00010
27	8.300	25.0	1.0	0.03002	0.04730	0.07188	0.00006
28	8.300	25.0	1.0	0.01089	0.01376	0.01507	-0.00122
29	8.300	25.0	1.0	0.00457	0.00261	0.00321	-0.00234
30	8.300	25.0	1.0	-0.00132	-0.00047	-0.00062	-0.00069
31	8.300	25.0	1.0	0.00149	0.00023	-0.00137	-0.00116
32	7.280	25.0	1.0	9.50076	4.78004	3.10086	0.00088
33	0.	25.0	1.0	0.00060	0.00158	0.00078	-0.00141

0	1	0.00000	13.70000	13.70000	100.00000	100.00000
0	2	0.00000	13.70000	13.70000	100.00000	100.00000
0	3	0.00006	13.70000	13.70000	100.00000	100.00000
0	4	0.00322	13.70000	13.70000	100.00000	100.00000
0	5	0.15890	13.70000	13.70000	100.00000	100.00000
0	6	7.21633	13.70000	13.70000	100.00000	100.00000
0	8	7.92722	13.70000	13.70000	100.00000	100.00000
0	9	2.09803	13.70000	13.70000	100.00000	100.00000
0	10	0.72900	13.70000	13.70000	100.00000	100.00000
0	11	0.22228	13.70000	13.70000	100.00000	100.00000
0	12	0.05446	13.70000	13.70000	100.00000	100.00000
0	13	0.01110	13.70000	13.70000	100.00000	100.00000
0	14	0.00196	13.70000	13.70000	100.00000	100.00000
0	15	0.	0.10000	0.	0.	0.

$$$$$ STREAM	FLOW	TEMP	PRES	PROCESS VARIABLES AT TIME = 4.00000E+00 COMPONENTS			$$$$$
1	7.280	25.0	1.0	9.50000	4.78000	3.10000	0.
2	2.430	25.0	1.0	-0.00030	-0.00287	0.00027	0.00083
3	2.430	25.0	1.0	-0.00171	-0.00010	0.00167	0.00104
4	2.430	25.0	1.0	0.00033	0.00152	-0.00112	-0.00119
5	2.430	25.0	1.0	0.00073	0.00142	0.00143	-0.00148
6	2.430	25.0	1.0	0.00517	0.00699	0.01112	0.00028
7	2.430	25.0	1.0	0.02658	0.03648	0.04882	0.00147
8	2.430	25.0	1.0	0.12005	0.16363	0.15260	0.00193
9	2.430	25.0	1.0	0.99189	0.88592	0.32952	-0.00105
10	2.430	25.0	1.0	13.35762	5.73408	0.44555	-0.00063
11	2.430	25.0	1.0	1.74371	0.38242	0.02164	-0.00014
12	2.430	25.0	1.0	0.03875	0.00932	0.00053	0.00060

Fig. 9. (Continued)

13	2.430	25.0	1.0	0.00248	-0.00026	0.00081	0.00062
14	2.430	25.0	1.0	0.00101	0.00035	-0.00022	-0.00116
15	2.430	25.0	1.0	0.00065	-0.00031	0.00017	-0.00085
16	2.430	25.0	1.0	0.00112	-0.00008	0.00103	-0.00036
17	1.020	25.0	1.0	0.00049	0.00080	0.00016	-0.00085
18	1.020	25.0	1.0	0.00074	-0.00050	0.00032	-0.00065
19	1.020	25.0	1.0	-0.00197	0.00111	-0.00088	-0.00036
20	1.020	25.0	1.0	0.00221	0.00112	0.00153	-0.00117
21	1.020	25.0	1.0	0.00563	0.00301	-0.00052	-0.00003
22	1.020	25.0	1.0	0.26645	0.06481	0.00526	0.00053
23	1.020	25.0	1.0	2.67681	1.47261	0.15891	0.00095
24	8.300	25.0	1.0	0.79757	0.92524	2.33062	-0.00209
25	8.300	25.0	1.0	0.08294	0.22279	1.55268	0.00043
26	8.300	25.0	1.0	0.02283	0.10186	0.84811	0.00005
27	8.300	25.0	1.0	0.01652	0.06668	0.40670	0.00013
28	8.300	25.0	1.0	0.01232	0.04621	0.17557	-0.00230
29	8.300	25.0	1.0	0.01111	0.02732	0.06770	-0.00021
30	8.300	25.0	1.0	0.00533	0.01115	0.02404	-0.00039
31	8.300	25.0	1.0	0.00391	0.00390	0.00846	-0.00033
32	7.280	25.0	1.0	9.50028	4.78047	3.10133	0.00110
33	0.	25.0	1.0	0.00156	0.00090	-0.00118	0.00029

0	1	0.00000	13.70000	13.70000	100.00000	100.00000
0	2	0.00001	13.70000	13.70000	100.00000	100.00000
0	3	0.00060	13.70000	13.70000	100.00000	100.00000
0	4	0.02820	13.70000	13.70000	100.00000	100.00000
0	5	1.30873	13.70000	13.70000	100.00000	100.00000
0	6	29.06158	13.70000	13.70000	100.00000	100.00000
0	8	16.46328	13.70000	13.70000	100.00000	100.00000
0	9	3.82766	13.70000	13.70000	100.00000	100.00000
0	10	1.47743	13.70000	13.70000	100.00000	100.00000
0	11	0.76358	13.70000	13.70000	100.00000	100.00000
0	12	0.40881	13.70000	13.70000	100.00000	100.00000
0	13	0.19195	13.70000	13.70000	100.00000	100.00000
0	14	0.07620	13.70000	13.70000	100.00000	100.00000
0	15	0.	0.10000	0.	0.	0.

Fig. 9. (Continued)

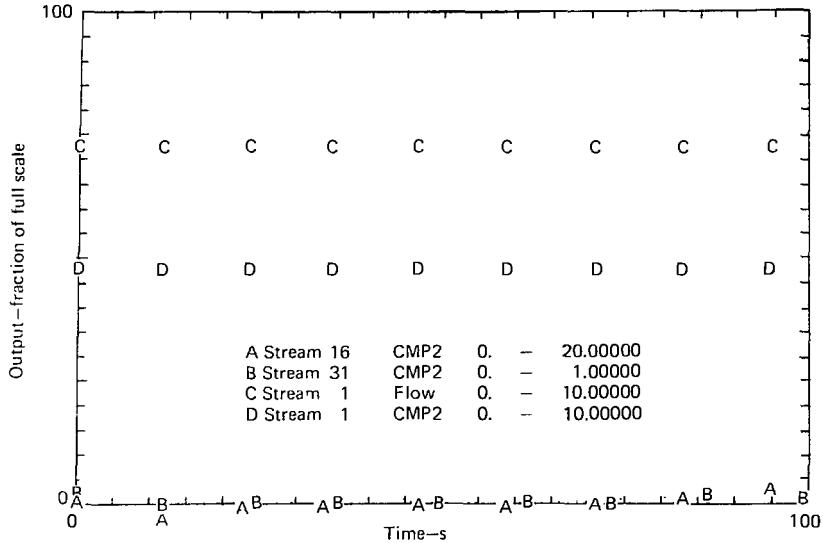


Fig. 10. DD80 graphical output of four selected variables from the 14-stage extractor simulation (100 s).

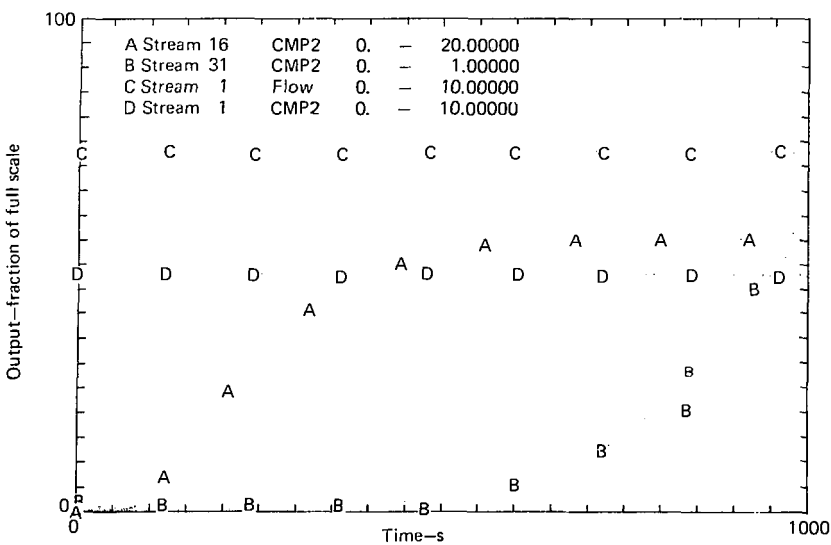


Fig. 11. DD80 graphical output of four selected variables from the 14-stage extractor simulation (1000 s).

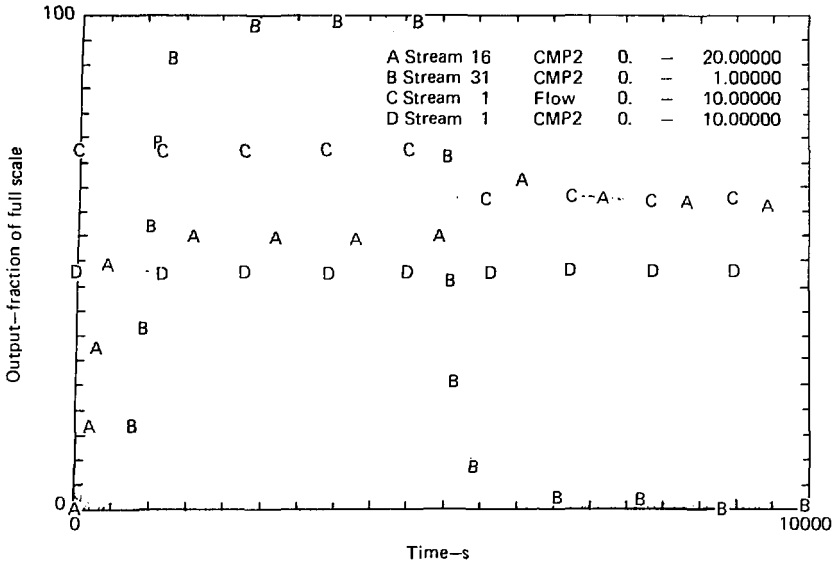


Fig. 12. DD80 graphical output of four selected variables from the 14-stage extractor simulation (10,000 s).

PROGRAM DEFICIENCIES

The DYNSYL code is now adequate for the dynamic simulation of a short series of units in a chemical plant, particularly one that is composed specifically of the unit modules now available for DYNSYL. The code has several shortcomings, however, that need to be remedied to ensure its most effective use. Some of these are summarized as follows:

- If some units are to be coupled and others not, at least two integrator programs are now required, because the GEAR-integrator is not written for such mixed operation.
- The code produces no flowsheet schematic in its output to key into the numerical and graphic output. Such a schematic would aid greatly in providing permanent documentation of the results.
- The graphic and interactive parts of the code are specific to the I.I.I. computer system and must be rewritten for other computer systems.
- For general use in the chemical industry many more unit module subprograms must be written. The new modules should be written to be as general as possible with specific applications determined by attached subroutines.

DYNSYL has already proven useful in producing simulated dynamic data for evaluation of on-line material control methods. Further applications of the code will gradually expand its usefulness.

APPENDIX A
UNIT MODULE SUBPROGRAMS

STGIDL (Type 1)

This subprogram as presently written represents one ideal stage for the coextraction of U^{+6} and Pu^{+4} from other salts. The stage may have two feeds to and two effluents from each phase—organic and aqueous. The volumes of the two phases are constant at predetermined values. The sum of volumetric input flow rates must, therefore, match the sum of volumetric output flow rates. Instantaneous output flow rates are made to match any input flow rate changes. If the aqueous volume is specified as zero, the module functions as a single-phase, perfectly mixed stage. As structured, the module may very easily be generalized to accommodate chemical reactions and, therefore, separation of Pu^{+3} and U^{+6}. Also, with a change in ORGPH, the equilibrium program and some statements containing the U and Pu molecular weights, any liquid-liquid extraction may be simulated.

The nomenclature used in the program is as follows:

VOL(1)	Organic phase volume (l)
VOL(2)	Aqueous phase volume (l)
XTBP	Mass fraction tributyl phosphate in organic solvent
KV	Rate constant for approach to equilibrium
KEXT	Rate constant for volume change (not used)
INSTGO, INFDO	Organic input streams (1,2)
OUTPRO, OUTSTO	Organic output streams (3,4)
INSTGA, INFDA	Aqueous input streams (5,6)
OUTSTA, OUTPRA	Aqueous output streams (7,8)
W(J)	Volumetric flow rate for stream J
T(J)	Temperature for stream J
P(J)	Pressure (not a variable) for stream J
LIN	Total input aqueous flow rate
VIN	Total input organic flow rate
LOUT	Total output aqueous flow rate
VOUT	Total output organic flow rate
XIN(I)	Average input aqueous concentration of component I
XOUT(I)	Average output aqueous concentration of component I
YIN(I)	Average input organic concentration of component I
YOUT(I)	Average output organic concentration of component I
TIN	Average input temperature
TOUT	Average output temperature
YIDEAL(I)	Equilibrium organic concentration of component I
DERY(I)	Rate of change of Y(I)
DERX(I)	Rate of change of X(I)
DISCO(I)	Ratio of YIDEAL (I) to X(I)

The equipment parameters, EP(IM,J), are as follows:

EP(IM,1)	Plutonium holdup, grams
EP(IM,2)	Organic phase volume (l)
EP(IM,3)	Aqueous phase volume (l)
EP(IM,4)	Rate constant for volume change
EP(IM,5)	Rate constant for approach to equilibrium
EP(IM,6)	Weight fraction tributyl phosphate in organic phase

CONTLR (Type 2)

This module is a simple algebraic proportional-plus-integral control element. Its input stream X is information from a stream (temperature, concentration, flow rate). The output is a signal to a control device, such as a valve (VALV). If the deviation from a set point is E = (X-set point)/range, the output signal Y is given by $dy/dt = K_p(dE/dt + K_I E)$, where K_p and K_I are proportional gain and integral constant, respectively. The module simulates this equation without using an outside integrator.

The nomenclature used in the module is as follows:

P1	Proportional gain
P2	Integral constant
IN	Input stream
OUT	Output signal
K	Variable index
OLD	Old value of variable
ERR	Deviation from set point
OLDER	Old deviation

The equipment parameters to be specified are:

EP(IM,1)	Controlled variable index
EP(IM,2)	Range of controlled variable
EP(IM,3)	Set point of controlled variable
EP(IM,4)	Proportional gain
EP(IM,5)	Integral constant

PIPE (Type 3)

Simulation of flow through a pipe is, for purposes of dynamic plant modeling, essentially time-delay with dispersion. If the pipe is relatively long, the dispersion may be ignored, so this module is a pure delay. It is a modification of DLAY in the original DYNSYS program. The delay time may be fixed or based on pipe length and fluid velocity. The delay is accomplished by storing values of the stream variables for the correct number of time increments, and then using the values previously stored as input (exit from the pipe). It is possible to delay only a selected part of a stream, making a crude simulation of dispersion possible. No integration is involved in this module.

The nomenclature is as follows:

SX(I,J,K)	Matrix of delayed stream values
I	Index of past times in SX
J	Maintained as 1 in SX
K	Index of stream variables
IN	Input stream (output from upstream unit)
OUT	Output stream (input to downstream unit)
TLAG	Value of delay time
BYP	Fraction of stream not delayed
NV	Number of storage spaces in delay vector
HH	Time increment

The equipment parameters that must be specified are:

EP(IM,1)	Value of a fixed delay time or negative of pipe volume for variable delay time.
EP(IM,2)	Fraction stream not delayed.
EP(IM,3)	Number of storage spaces in delay vector.
EP(IM,4)	Flag: zero (flow rate may vary), one (flow rate remains constant at input value).

EXTRTR (Type 5)*

This is a module for multistage extraction and chemical reaction simulation. It is based on a program written by L. E. Burkhart of Iowa State University, and is capable of simulating plutonium-uranium separation or coextraction with variable phase volumes. The module does not account for any backmixing or dispersion effects. The number of equilibrium stages may be specified up to 20; the feed stage may also be specified. The stage volumes are uniform and are determined from the specified column volume. It is also possible to specify the time constant for changes in the phase volume ratio. Integration occurs within this module, so the integrator portions of DYNSYL are not used.

A schematic of the extraction column modeled is shown in Fig. A-1 with names of some of the variables shown. The nomenclature used in the module is as follows:

AQF	Aqueous feed rate (l/s).
AP	Pseudo aqueous phase flow rate (l/s).
AQ	Actual aqueous phase flow rate (does not vary from stage to stage).
A1 thru A9	Constants used in calculating uranium(VI) and plutonium(IV) distribution coefficients.
B1 thru B3	Constants used in calculating distribution coefficients for NO_3.
DDTAQ	Time derivative of pseudo aqueous phase flow rate.
DDTHU	Time derivative of aqueous phase holdup.
DDTOR	Time derivative of pseudo organic phase flow rate.
DDTORF	Time derivative of organic feed flow rate.
DT	Time increment between integration steps (s).
DQ	New aqueous flow rate computed in control subrouting (l/s).
EH2	Distribution coefficient for nitrous acid.
EH3	Distribution coefficient for nitric acid.
EMH2	$(Or/H)E_{HNO_2}$.
EMH3	$(Or/H)E_{HNO_3}$.
EMP	$(Or/H)E_{Pu(IV)}$.
EMU	$(Or/H)E_{U(VI)}$.
ENH2	$(Or'/Aq')E_{HNO_2}$.
ENH3	$(Or'/Aq')E_{HNO_3}$.
ENP	$(Or'/Aq')E_{Pu(IV)}$.
ENU	$(Or'/Aq')E_{U(VI)}$.
EOAP	$E_{Pu(IV)}$.
EOAU	$E_{U(VI)}$.
FIS	$10^{0.91\mu_i^{1/2}} - 1.521$.
FTBP	C'', free TBP concentration used in calculating the distribution coefficient for HNO_3.
HNO2AZ	$[HNO_2]$, aqueous feed concentration (moles/l).
HNAZ	[HN], aqueous feed concentration (moles/l).
HZAZ	[HZ], aqueous feed concentration (moles/l).
H	$[H^+]$, hydrogen ion concentration, aqueous, (moles/l).
HF	$[H^+]$ at time t + Δt.
HN	$[NH_3OH^+]$, hydroxylamine concentration, aqueous (moles/l).
HNF	$[NH_3OH^+]$, at time t + Δt.
HNO2	$[HNO_2]$, nitrous acid concentration, aqueous (moles/l).
HNO2F	$[HNO_2]$, at time t + Δt.
HZ	$[N_2H_5^+]$, hydrazine concentration, aqueous (moles/l).
HZF	$[N_2H_5^+]$ at time t + Δt.
HNO2Z	HNO_2 feed concentration in organic feed (moles/l).
HNO2AZ	HNO_2 feed concentration in aqueous feed (mols/l).
HU	Aqueous phase holdup (l).
HOLDUP	Name for subroutine for calculating holdups and flow rate parameters.

*There is no Type 4.

KH	K_H, ionic strength function for $[H^+]$.
KP	$K_{Pu(IV)}$
KU	$K_{U(VI)}$
K1	Rate constant for reaction 1.
K2	Constant used in TEM4 in material balance calculations.
K3	Rate constant for reaction 3.
K4	Rate constant for reaction 2.
K5	Rate constant for reaction 4.
L	Number of equilibrium stages in the column.
M	Symbol for fictitious stage (L+1), used to specify inputs and outputs at the bottom stage (L).
N	Number of iterations or time steps specified.
NF	Feed stage number.
NO3	NO_3 concentration in aqueous phase (moles/l).
NO3O	NO_3 concentration in organic phase (moles/l).
NO3I	$[NO_3]$ at time t + Δt, aqueous.
NO3Z	New NO_3 concentration in organic feed (moles/l).
NO3AZ	NO_3 concentration in the aqueous feed (moles/l).
N7	(NF -1), stage before feed stage.
N8	(NF +1), stage after feed stage.
OP	Pseudo organic phase flow rate.
OR	Organic phase flow rate (does not vary from stage to stage) (l/s).
ORI	Organic feed flow rate immediately preceding a flow rate upset (l/s).
OR1	Organic phase flow rate in scrub section (l/s).
OR2	Organic phase flow rate in extraction section (OR2 = OR1 + ORF).
P3A	Pu^{+3} concentration in aqueous phase (moles/l).
P3AI	Pu^{+3} at time t + Δt, aqueous.
P4A	Pu^{+4} concentration in aqueous phase (moles/l).
P4AI	Pu^{+4} at time t + Δt.
P4O	Pu^{+4} concentration in organic phase (moles/l).
P4OZ	Pu^{+4} concentration in organic feed (moles/l).
P4OAZ	Pu^{+4} concentration in organic feed immediately following a Pu feed concentration upset (moles/l).
PUBAL	Residual from overall plutonium material balance over the column.
RN	-(Aq +H).
RXN1 thru RXN5	Incremental change in material gained or lost to reaction for each time step as determined by the kinetic rate equations.
TAQ	τ_{Aq}, interstage time constant for pseudo aqueous flow rate (s -1).
TOR	τ_{Or}, interstage time constant for pseudo organic flow rate (s -1).
UOZ	$[U(VI)]_{Or}$, organic feed concentration (moles/l).
UAZ	$[U(VI)]_{Aq}$, aqueous feed concentration (moles/l).
UA	$[U(VI)]_{Aq}$, aqueous feed concentration (moles/l).
UAI	$[U(VI)]_{Aq}$, at time t + Δt.
UO	$[U(VI)]_{Or}$, U^{+4} concentration in the organic phase (moles/l).
UOF	Uranium concentration in feed at steady-state immediately preceding an upset in uranium feed concentration (moles/l).
VOL	Volume of a single theoretical stage (l).

The equipment parameters required by this module are:

EP(IM,1)	Plutonium holdup in column, g
EP(IM,2)	Number of equilibrium stages
EP(IM,3)	Feed stage
EP(IM,4)	Column volume (l)
EP(IM,5)	Time constant for phase volume change (s)
EP(IM,6)	Weight fraction tributyl phosphate in organic stream

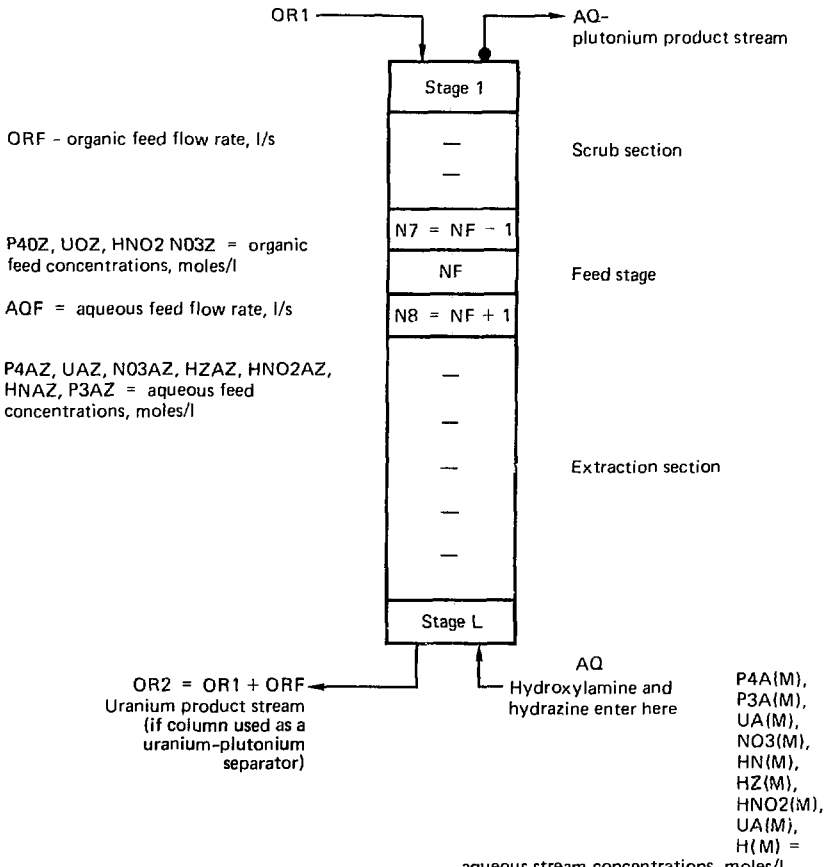


Fig. A-1. Extraction column showing location of streams and principal variables used in computer program.

PRECIP (Type 6)

This module simulates the dynamic behavior of a precipitator-reactor for the precipitation of plutonium as oxalate. A major component of the simulation is involved in the computation of nucleation and growth rates of the precipitated $Pu(C_2O_4)_2$ particles. Reference 15 details the derivation of the equations used to model the particle nucleation and growth process in the reactor. To use the module it is only necessary to specify the reactor volume as the nucleation and growth rate constants are specified in the program. Two streams must enter the reactor, one containing the plutonium solution, the other the oxalate solution. Only one stream exits the reactor—a mixture of filtrate and precipitate.

The nomenclature used in the module is as follows:

AKG, ANG, AKB, ANB, RZRO	Constants in nucleation and growth equations
RO	$Pu(C_2O_4)_2$ particle density (g/cc)
V	Reactor volume (l)
Y(1)	Concentration of plutonium in reactor solution (filtrate), (g/l)
Y(2)	Filtrate flow rate (l/h)
Y(3)	Concentration of NO_3^- in reactor solution (moles/l)
Y(4)	Concentration of $C_2O_4^=$ in reactor (moles/l)
Y(5)	Zero moment of distribution
Y(6)	First moment of distribution
Y(7)	Second moment of distribution
Y(8)	Third moment of distribution
FP	Plutonium-solution feed rate (l/h)
FO	Oxalate-solution feed rate (l/h)
CP	Concentration of plutonium in feed stream (g/l)
CO	Concentration of oxalate in oxalate-feed stream (moles/l)
TFR	Total feed rate (l/h)

The equipment parameters are:

EP(IM,1)	Plutonium holdup (g)
EP(IM,2)	Reactor volume (l)

The output variables are as follows:

S(2,NOUT,6)	Rate of change of feed flow rates
S(2,NOUT,7)	Rate of change of plutonium concentration in filtrate
S(2,NOUT,8)	Rate of change of NO_3^- concentration in filtrate
S(2,NOUT,9)	Rate of change of $C_2O_4^=$ concentration in filtrate
S(2,NOUT,10-12)	Rates of change of particle size distribution moments

EVAPTR (Type 7)

This module is the simplest possible version of an evaporator module that may be used as a plutonium-solution concentrator simulation. It is an adaptation of the perfectly-mixed evaporator written by Rudolf Rozsa at LLL. The solution properties were all internally generated and were fixed in the earlier program. This program was generalized to use variable properties (density, heat of vaporization, heat capacity) computed by the property subroutines in DYNSYL. We integrated analytically in this module because the balance equations were linear.

The nomenclature used is as follows:

IN	Input stream
OUT	Output stream
VAP	Vapor stream
STEAM	Steam stream to heater
MFIN	Feed rate (moles/h)
XIPU	Plutonium mole fraction in feed
MSTM	Steam rate (moles/h)
PVPS	Vapor presure
PRESS	Presure in vapor stream
TEMP	Temperature of vapor stream
DAVGOUT	Density of product liquid
XOPU	Plutonium mole fraction in product
V	Evaporator volume (liquid phase)

The only equipment parameters are:

EP(IM,1)	Plutonium holdup (g)
EP(IM,2)	Volume (l)
CPi	Heat capacity of stream –i
RHOi	Density of stream –i

GNTRNS (Type 8)

This is a simulation of a unit capable of heat transfer, two-phase equilibrium (liquid-liquid or vapor-liquid), and chemical reaction. The module is not yet perfected and more changes will undoubtedly be made. When perfected, the module will be able to simulate the dynamic behavior of ideal extraction and distillation (or flash) stages, mixed reactors, evaporators, heat exchangers, or condensers.

The nomenclature used in GNTRNS is the same as in STGIDL with some additional terms. These are as follows:

HCOEF	External heat transfer coefficient
HAREA	External heat transfer area
NTYPE	Zero for extraction; one for flash
HT etc.	Heat capacity

The equipment parameters to be specified are as follows:

EP(IM,1)	Plutonium holdup (g)
EP(IM,2)	Total volume (l)
EP(IM,3)	Aqueous or liquid phase volume (l)
EP(IM,4)	Rate constant for volume change
EP(IM,5)	Rate constant for approach to equilibrium
EP(IM,6)	Volume fraction tributyl phosphate
EP(IM,7)	External heat transfer coefficient [cal/(cm ^{2}·s·°C)]
EP(IM,8)	External heat transfer area (cm ^{2})
EP(IM,9)	Stage type: 0 for extraction, 1 for flash

VALV (Type 9)

As written, this module simulates the simplest possible control valve or electrical current controller. The output rate is simply proportional to the controller signal or proportional to the maximum controller signal (1.0 in this case) minus the current signal. Generalization to include other actions would be very simple.

The nomenclature is:

IN	Input signal from controller (CONTLR)
OUT	Controlled stream flow rate (or current if electrical)

The unit parameters are:

EP(IM,1)	Proportionally constant for valve action.
EP(IM,2)	Positive causes proportional action; negative causes reverse proportional action.

APPENDIX B
PROGRAM LISTING WITH UNIT MODULE SUBPROGRAMS

The DYNSYL code has many comments that help explain each section to the user. We have endeavored to make the unit module subprogram even more self-explanatory. The code is listed with the primary sections in the following order: main and input-output, physical properties, integrator, and unit modules. The order of the specific subroutines and their purpose is as follows:

MAIN	Calling program
DYN1	Numerical input and initialization
GET	Alphameric input
DYN2	Unit module calling program
OUTPUTS	Numerical output of state variables
SAVEP	Computation of plot variables
WRITEP	Output of measurement vector
LPLOT	Lineplot output
CALPLT	DI3800 plot routine
PROPS	Read constants for physical properties
MOWT	Average molecular weight
CPLA	Average liquid heat capacity
CPVA	Average vapor heat capacity
ENTL	Average liquid enthalpy
ENTV	Average vapor enthalpy
LAMB	Heat of vaporization
DENL	Average liquid molal density
DENV	Average vapor molal density
VAPR	Vapor pressure
WILS	Activity coefficients
BUBL	Bubble point
KVAL	Vapor-liquid equilibrium ratio
TEMPL	Liquid heat balance
TEMPV	Vapor heat balance
DRIVE	Main program of integrator
NOISE	Process and measurement noise
STIFF	Integration control program
PSET	Computation with Jacobian
INTERP	Interpolation of state variables
COSET	Integration coefficients
DEC	Matrix triangularization
SOL	Solution matrix equations
STGIDL	General ideal stage
EXTRTR	Multistage extraction for uranium and plutonium
PRECIP	Precipitator for plutonium oxalate
EVAPTR	Evaporative concentrator for $Pu(NO_3)_4$
CONTLR	Controller
PIPE	Flow delay and dispersion
PUMP	Flow driver and head source

```
144 10    CONTINUE
145 C
146 C          INITIALIZE PRINT AND INTEGRATOR PARAMETERS
147 C
148       NPRINT=0
149       INDEX=1
150       JSTART=0
151 C
152 C          READ DATA
153 C
154       CALL DYN1
155 C
156 C     SET UP INITIAL VALUES FOR GRAPH VECTOR COUNT
157 C
158       NPTS=0
159       PLOTT=0.0
160 20    CONTINUE
161 C
162 C          TERMINAL INPUT
163 C
164       WRT 59,"(""TIME IS NOW "",F10.5)",TIME
165       WRT 59,"(""TYPE TIME FOR NEXT PROCESS CHANGE, F10.5"")"
166       RIT 59,"(F10.5)",CTIME
167 C
168 C          COMPUTATION INTERVAL SIZE
169 C
170       HINC=TMAX/NPR
171       HH=HINC/10.0
172       OTIME=TIME
173 C
174 C          INITIAL PLOT AND PRINTOUT VALUES
175 C
176       IF (NSAVE.GT.0) CALL SAVEP
177       IF (NOUTPT.EQ.1) CALL OUTPUTS
178       JSTART=1
179       IF (TIME.NE.0.0)  GO TO 40
180 C
181 C          INITIAL MEASUREMENT DATA POINTS
182 C
183       GO TO 41
184 40    CONTINUE
185       TIME=OTIME+HINC
186       IF (TIME.LT.CTIME) GO TO 41
187 C
188 C          TERMINAL INPUT
189 C
190       WRT 59,"(""TIME IS NOW "",F10.5)",TIME
191       WRT 59,"(""TYPE NUMBER OF COMPONENTS CHANGED, I5"")"
192       RIT 59,"(I5)",NUM
193       DO 46 I=1,NUM
194       WRT 59,"(""TYPE CHANGE STREAM AND COMP. NUMBERS, 2I5"")"
195       RIT 59,"(2I5)",IC,JC
196       WRT 59,"(""TYPE NEW VALUE FOR S(I,J), F10.5"")"
197       RIT 59,"(F10.5)",S(I,IC,JC)
198    46 CONTINUE
199       WRT 59,"(""TYPE TIME FOR NEXT PROCESS CHANGE, F10.5"")"
200       RIT 59,"(F10.5)",CTIME
201    41 CONTINUE
202       INTFL=0
203       IF(N0.GE.NF) GO TO 30
204       N=0
205 C
206 C          CONVERT VARIABLE MATRIX TO VECTOR FOR INTEGRATOR
207 C
208       DO 43 I=N0,N5
209       DO 42 J=N1,N2
210       N=N+1
211       Y(N)=S(I,I,J)
212    42 CONTINUE
213    43 CONTINUE
214 C
215 C          CALL TO INTEGRATOR SUBROUTINE
```

```
C               SEE NOTES IN DRIVE FOR ARGUMENT VARIABLE DEFINITIONS
C               AND VALUES FOR IMETH, THE INTEGRATION METHOD PARAMETER
C
      CALL DRIVE(N,DTIME,H,Y,TIME,EPS,IMETH,INDEX)
      N=0
C
C           CONVERT VECTOR BACK TO VARIABLE MATRIX
C
      DO 45 I=NB,NF
      DO 44 J=N1,N2
      N=N+1
      S(1,I,J)=Y(N)
   44 CONTINUE
   45 CONTINUE
C
C           COMPUTATIONS WITHOUT USE OF DRIVE
C
   30 IF (NF.EQ.NS) GO TO 39
      INTFL=1
      T=DTIME
      Y(1)=0.0
      YDOT(1)=0.0
      N=1
      DO 38 I=1,10
      T=T+HH
      CALL DYN2(N,T,Y,YDOT)
   38 CONTINUE
   39 CONTINUE
C           SAVE VALUES FOR PLOT
C
      IF(NSAVE.GT.0) CALL SAVEP
      NPRINT=NPRINT+1
      DTIME=TIME
      IF (NPRINT.LT.NPRT) GO TO 77
      NPRINT=0
C
C           PRINT OUTPUT IF NPRINT = NPRT
C
      IF(NDUTPT.EQ.1) CALL OUTPUTS
   77 N=0
C
C           COMPUTE MEASUREMENT VECTOR
C
      DO 72 I=NB,NS
      DO 71 J=N1,N2
      N=N+1
      XPLOT(N)=S(1,I,J)
   71 CONTINUE
   72 CONTINUE
      NSTATE=N
C
C           ADD MEASUREMENT NOISE
C
      CALL NOISE (NSTATE,XPLOT,EMSTD,EMPRCT,EMC)
C
C           PRINT MEASUREMENT DATA
C
      IF (NNUM0.EQ.1)  CALL WRITEP
   47 CONTINUE
      TFIN=0.99999*TMAX
      IF (TIME.LT.TFIN) GO TO 40
C
C           PRINT PRINTPLOT
C
      IF (NLINE.EQ.1) CALL LPLOT
C
C           PLOTTER CALL
C
      IF (NCAL.EQ.1) CALL CALPLT
C
C           TEST FOR CONTINUATION DATA
C
      CALL GET (N,M,X,0)
```

```
      IF (N.EQ.4HEND ) CALL EXIT(1)
C
C          REPEAT SAME SIMULATION
C
      IF (N.EQ.4HREPE) GO TO 10
C
C          CONTINUE FOR ADDITIONAL TIME
C
      IF (N.EQ.4HCONT) GO TO 100
      WRITE (NERR,110)
      CALL EXIT(1)
100   CALL DYN1
      GO TO 20
C
C
110   FORMAT (41H LAST CARD MUST BE END,REPEAT OR CONTINUE)
120   FORMAT (/,20X,21H+++TMAX APPROACHED+++)
      END
C
C
      SUBROUTINE DYN1
C
C     DYN1 READS AND ECHOES THE DATA SET
C
      COMMON /UNIT/ IM,NMP
      COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
      COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,N8,N1,N2,NF
      COMMON /PLT/ NPLOTS,PLOTI,PLOTD(15,4),PLOTT,PTYPE
      COMMON /OUT/ NOUTPT,NLINE,NCAL,NPR,NSAVE,NNUMO
      COMMON /GERR/ JSTART,IMETH,TIME,H,HH,HINC,EPS,T,INTFL
      COMMON /IO/ NIN,NOUT,NERR,NPOINT,NPRT
      COMMON/LARM/STD,PRCNT,EMC,EMSTD,EMPRCT
      DIMENSION AM(200),SS(2,45,11)
      DIMENSION ITAG(30), JTAG(30)
C
C          EQUIPMENT (UNIT)  MODULE NAMES
C
      DATA ITAG/4HSTGI,4HCONT,4HPIPE,4HPUMP,4HEXTR,4HPREC
     1,4HEVAP,4HGNTR,4HVALV,21*4H    /
      DATA JTAG/4HDL  ,4HLR  ,4H    ,4H    ,4HTR  ,4HIP
     1,4HTR  ,4HNS  ,22*4H    /
C
      IF (JSTART.NE.0) GO TO 310
      WRITE (NOUT,390)
C
C          DEFAULT VALUES
C
      EMC=0.0
      NIST=1
      NFIN=6
      N8=1
      NF=0
      MAXNE=35
      TIME=0.0
      ISTIFF=2
      MITER=2
      H=1.0E-08
      NCOMP=1
      NMP=5
      TMAX=10.0
      NPR=1
      EPS=0.001
      NOUTPT=0
      NNUMO=0
      NLINC=0
      NCAL=0
      NOMES=0
      XMIN=1.0E-06
      STD=0.0
      PRCNT=0.0
      EMSTD=0.0
      EMPRCT=0.0
      NPRT=1
```

```
C
C           READ AND COPY TITLE
C
10      READ (NIN,410) (AM(I),I=1,18)
        WRITE (NOUT,400) (AM(I),I=1,18)
        IF (AM(1).NE.4HBEGI) GO TO 10
C
C       READ SIMULATION DATA
C
20      CALL GET (N,M,X,1)
        IF (N.EQ.4HNONS) ISTIFF=1
        IF (N.EQ.4HCOMP) NCOMP=X+0.05
        IF (N.EQ.4HIN/O) NMP=X+0.05
        IF (N.EQ.4HFEED) NB=X+0.05
        IF (N.EQ.4HCOUP) NF=X+0.05
        IF (N.EQ.4HFIRS) NIST=X+.05
        IF (N.EQ.4HLAST) NFIN=X+.05
        IF (N.EQ.4HTIME) TMAX=X
        IF (N.EQ.4HNO O) NPR=X+0.05
        IF (N.EQ.4HTOLE) EPS=X
        IF (N.EQ.4HITER) NPRT=X+0.05
        IF (N.EQ.4HNDIS) STD=X
        IF (N.EQ.4HPERC) PRCNT=X
        IF (N.EQ.4HMEAS) EMSTD=X
        IF (N.EQ.4HM NO) EMPRCT=X
        IF (N.EQ.4HOUTP) NOUTPT=1
        IF (N.EQ.4HNUMO) NNUMO=1
        IF (N.EQ.4HLINE) NLINE=1
        IF (N.EQ.4HCALP) NCAL=1
        IF (N.EQ.4HNOME) NOMES=1
        IF (N.EQ.4HMINP) XMIN=X
        IF (N.EQ.4HINTE) MITER=X+0.05
        IF (N.EQ.4HDELT) H=X
        IF (N.EQ.4HLIBR) GO TO 30
        IF (N.EQ.4HPROC) GO TO 50
        GO TO 20
C
C           READ NEW UNIT MODULE NAME
C
30      NLIB=X+0.05
        DO 40 I=1,NLIB
        CALL GET (N,M,X,0)
        J=X+0.05
        ITAG(J)=N
        JTAG(J)=M
40      CONTINUE
        GO TO 20
50      CONTINUE
C
C       READ EQUIPMENT DATA
C
        IMETH=10*ISTIFF+MITER
        N1=NIST+2
        N2=NFIN+2
        NB=NB+1
        NMP=NMP+2
        MAXNMP=NMP+1
        NC3=NCOMP+3
        NC5=NCOMP+5
        DO 80 I=1,MAXNE
        DO 60 J=1,MAXNMP
        MP(I,J)=0
60      CONTINUE
        DO 70 J=1,10
        EP(I,J)=0.0
70      CONTINUE
80      CONTINUE
        NE=0
        NEX=1
90      CALL GET (N,M,X,0)
100     IF (N.EQ.4HEND ) GO TO 140
        DO 110 I=1,30
        IF (N.EQ.ITAG(I).AND.M.EQ.JTAG(I)) GO TO 120
```

```
435 110    CONTINUE
436        WRITE (NERR,420) N,M
437        CALL EXIT(1)
438 120    CONTINUE
439        NE=NE+1
440        MP(NE,1)=X+SIGN(0.05,X)
441        MP(NE,2)=I
442        READ (NIN,450) (AM(I),I=3,NMP)
443        DO 130 J=3,NMP
444        X=AM(J)
445        MP(NE,J)=X+SIGN(0.05,X)
446 130    CONTINUE
447        WRITE (NOUT,430) (MP(NE,I),I=1,2)
448        WRITE (NOUT,470) (AM(I),I=3,NMP)
449        READ (NIN,450) (EP(NE,I),I=1,10)
450        WRITE (NOUT,470) (EP(NE,I),I=1,10)
451        CALL GET (N,M,X,0)
452        IF (N.NE.4HEXTR) GO TO 100
453        NNX=X+0.05
454        NMX=NNX+NEX-1
455        READ (NIN,450) (EX(J),J=NEX,NMX)
456        WRITE (NOUT,450) (EX(J),J=NEX,NMX)
457        MP(NE,NMP+1)=NEX
458        NEX=NEX+NNX
459        GO TO 90
460 C
461 C      READ STREAM DATA
462 C
463 140    CALL GET (N,M,X,0)
464        IF (N.NE.4HSTRE) GO TO 150
465        GO TO 160
466 150    WRITE (NERR,360)
467        CALL EXIT(1)
468 160    CONTINUE
469        NS=X+0.05
470        DO 180 I=1,NS
471        S(1,I,1)=I
472 C
473 C           DEFAULT STREAM FLAG
474 C
475        S(1,I,2)=1.0
476 C
477 C           DEFAULT FLOW RATE, L/S
478 C
479        S(1,I,3)=0.0
480 C
481 C           DEFAULT TEMPERATURE, C
482 C
483        S(1,I,4)=25.0
484 C
485 C           DEFAULT PRESSURE, ATM
486 C
487        S(1,I,5)=1.0
488 C
489 C           DEFAULT CONCENTRATIONS
490 C
491        DO 170 J=6,NC5
492        S(1,I,J)=0.0
493   170  CONTINUE
494   180  CONTINUE
495 190    CALL GET (N,M,X,-1)
496        IF (N.EQ.4HEXPL) GO TO 200
497        IF (N.EQ.4HEND ) GO TO 250
498        IF (N.EQ.4HSPEC) GO TO 220
499        GO TO 190
500 C
501 C           READ STREAM VALUES FOR SPECIFIC STREAMS NUMBERED AM(1)
502 C
503 200    READ (NIN,440) IT,(AM(I),I=1,5)
504        IF (IT.EQ.4HEND ) GO TO 250
505        READ (NIN,450) (AM(I),I=6,NC5)
506        N=AM(1)+0.05
507        DO 210 I=1,NC5
508        S(1,N,I)=AM(I)
```

```
509 210   CONTINUE
510       GO TO 190
511 220   NN1=X+0.05
512 C
513 C          READ STREAM VALUES FOR STREAM #AM(1) INTO STREAMS X TO AM(1)
514 C
515       READ (NIN,450) (AM(I),I=1,NC5)
516       NN2=AM(1)+0.05
517       DO 240 I=NN1,NN2
518       DO 230 J=2,NC5
519       S(1,I,J)=AM(J)
520 230   CONTINUE
521       S(1,I,1)=I
522 240   CONTINUE
523       GO TO 190
524 250   DO 270 I=1,NS
525       WRITE (NOUT,460) (S(1,I,J),J=1,5)
526       WRITE (NOUT,470) (S(1,I,J),J=6,NC5)
527       DO 260 J=1,NC5
528       S(2,I,J)=0.0
529 260   CONTINUE
530 270   CONTINUE
531 C
532 C     GET PHYSICAL PROPERTIES DATA
533 C
534       CALL PROPS
535       NSAVE=NNUMD+NLINE+NCAL
536       IF (NSAVE.EQ.0) RETURN
537 C
538 C     READ GRAPHICAL DATA
539 C
540       CALL GET (N,M,X,0)
541       IF (N.NE.4HGRAP) GO TO 280
542       GO TO 290
543 280   WRITE (NERR,370)
544       CALL EXIT(1)
545 290   CONTINUE
546       NPLOTS=X+0.05
547       READ (NIN,450) PLOTI,PTYPE
548       WRITE (NOUT,470) PLOTI,PTYPE
549       DO 300 I=1,NPLOTS
550       READ (NIN,450) (PLOTD(I,J),J=1,4)
551       WRITE (NOUT,470) (PLOTD(I,J),J=1,4)
552 300   CONTINUE
553       RETURN
554 C
555 C     READ CONTINUATION DATA
556 C
557 310   CONTINUE
558       H=0.00000001
559       TMAX1=TMAX
560 320   CALL GET (N,M,X,1)
561       IF (N.EQ.4HTIME) TMAX=X
562       IF (N.EQ.4HNO D) NPR=X+0.05
563       IF (N.EQ.4HITER) NPRT=X+0.05
564       IF (N.EQ.4HTIME) GO TO 320
565       IF (N.EQ.4HNO D) GO TO 320
566       IF (N.EQ.4HITER) GO TO 320
567       HINC=(TMAX-TMAX1)/NPR
568       IF (N.EQ.4HSTRE) GO TO 330
569       IF (N.EQ.4HEQUI) GO TO 350
570       IF (N.EQ.4HEND ) RETURN
571       WRITE (NERR,380)
572       CALL EXIT(1)
573 330   NOS=X+0.05
574       READ (NIN,450) (SS(1,NOS,J),J=1,NC5)
575       DO 331 J=1,NC5
576       IF(SS(1,NOS,J).NE.0.0)  S(1,NOS,J)=SS(1,NOS,J)
577   331 CONTINUE
578       WRITE (NOUT,470) (S(1,NOS,J),J=1,NC5)
579       DO 340 J=1,NC5
580       S(2,NOS,J)=S(1,NOS,J)
581 340   CONTINUE
```

```
582         GO TO 320
583 350     NDEP=X+0.05
584         READ (NIN,450) (EP(NDEP,J),J=1,10)
585         WRITE (NOUT,470) (EP(NDEP,J),J=1,10)
586         GO TO 320
587 C
588 C
589 C       FREE FORMAT (DEC SYSTEM-10)
590 C
591 C   44 FORMAT(A4,5F)
592 C   45 FORMAT(5F)
593 C
594 360     FORMAT (27H STREAMS CARD SHOULD FOLLOW)
595 370     FORMAT (25H GRAPH CARD SHOULD FOLLOW)
596 380     FORMAT (27H ERROR IN CONTINUATION DATA)
597 390     FORMAT (1H1)
598 400     FORMAT (1X,10A4)
599 410     FORMAT (16A4)
600 420     FORMAT (16H EQUIPMENT NAME ,2A4,21H HAS NOT BEEN DEFINED)
601 430     FORMAT (5H UNIT,I3,6H   TYPE,I3)
602 440     FORMAT (A4,8X,5F12.0)
603 450     FORMAT (12X,5F12.5)
604 460     FORMAT (7H STREAM,5X,5F12.5)
605 470     FORMAT (12X,5F12.5)
606 490     FORMAT (/47HALL S(K,I,3) AND EP(IM,3 AND 4) ARE DIVIDED BY ,2X,F6.3)
607         END
608         SUBROUTINE GET (NAME1,NAME2,X,IFG)
609 C
610 C       GET IS CALLED BY DYN1 TO READ AND ECHO THE ALPHANUMERIC
611 C          DATA WORDS OF THE DATA SET
612 C
613         COMMON /IO/ NIN,NOUT,NERR,NPOINT
614 10      READ (NIN,60) NAME1,NAME2,NAME3,X
615         IF (NAME1.NE.4H    ) GO TO 20
616         IF (ABS(X).LT.1.0E-20) GO TO 10
617         WRITE (NERR,70)
618         GO TO 10
619 20      CONTINUE
620         IF (ABS(X).LT.1.0E-20) GO TO 30
621         IF (IFG) 30,40,50
622 30      WRITE (NOUT,90) NAME1,NAME2,NAME3
623         RETURN
624 40      N=X+SIGN(0.01,X)
625         WRITE (NOUT,80) NAME1,NAME2,NAME3,N
626         RETURN
627 50      WRITE (NOUT,90) NAME1,NAME2,NAME3,X
628         RETURN
629 C
630 C       FREE FORMAT (DEC SYSTEM-10)
631 C
632 C    6 FORMAT(3A4,5F)
633 C
634 C
635 60      FORMAT (3A4,F12.0)
636 70      FORMAT (46H WARNING - DATA SKIPPED WHILE READING KEYWORDS)
637 80      FORMAT (1X,3A4,I11)
638 90      FORMAT (1X,3A4,F11.5)
639         END
640         SUBROUTINE DYN2(N,T,Y,YDOT)
641 C
642 C       DYN2 IS CALLED BY THE INTEGRATOR PROGRAM (DRIVE)
643 C       DYN2 MARCHES DOWN THE PROCESS MATRIX, CALLING THE SUBROUTINES
644 C          REPRESENTING THE DIFFERENT UNIT COMPUTATIONS TO BE EXECUTED
645 C          ON BOTH THE PREDICTOR AND CORRECTOR STEPS
646 C
647         COMMON /UNIT/ IM,NMP
648         COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
649         COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
650         COMMON /GERR/ JSTART,IMETH,TIME,H,HINC,EPS,TT,INTFL
651         DIMENSION YDOT(N),Y(N)
652         IF (NB.GE.NF) GO TO 6
653         N=0
654 C
```

```
C           CONVERT VECTOR TO VARIABLE MATRIX
C
      DO 5 I=NB,NF
      DO 4 J=N1,N2
      N=N+1
      S(1,I,J)=Y(N)
    4 CONTINUE
    5 CONTINUE
      DO 320 IM=1,NE
      NTYPE=MP(IM,2)
      GO TO (10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,170,
     1180,190,200,210,220,230,240,250,260,270,280,290,300), NTYPE
   10 CALL TYPE1
      GO TO 310
   20 CALL TYPE2
      GO TO 310
   30 CALL TYPE3
      GO TO 310
   40 CALL TYPE4
      GO TO 310
   50 CALL TYPE5
      GO TO 310
   60 CALL TYPE6
      GO TO 310
   70 CALL TYPE7
      GO TO 310
   80 CALL TYPE8
      GO TO 310
   90 CALL TYPE9
      GO TO 310
  100 CALL TYPE10
      GO TO 310
  110 CALL TYPE11
      GO TO 310
  120 CALL TYPE12
      GO TO 310
  130 CALL TYPE13
      GO TO 310
  140 CALL TYPE14
      GO TO 310
  150 CALL TYPE15
      GO TO 310
  160 CALL TYPE16
      GO TO 310
  170 CALL TYPE17
      GO TO 310
  180 CALL TYPE18
      GO TO 310
  190 CALL TYPE19
      GO TO 310
  200 CALL TYPE20
      GO TO 310
  210 CALL TYPE21
      GO TO 310
  220 CALL TYPE22
      GO TO 310
  230 CALL TYPE23
      GO TO 310
  240 CALL TYPE24
      GO TO 310
  250 CALL TYPE25
      GO TO 310
  260 CALL TYPE26
      GO TO 310
  270 CALL TYPE27
      GO TO 310
  280 CALL TYPE28
      GO TO 310
  290 CALL TYPE29
      GO TO 310
  300 CALL TYPE30
  310 CONTINUE
```

```
728 320   CONTINUE
729       IF (NB.GE.NF) GO TO 342
730       N=0
731 C
732 C          CONVERT VARIABLE MATRIX TO VECTOR
733 C
734       DO 341 I=NB,NF
735       DO 340 J=N1,N2
736       N=N+1
737       YDOT(N)=S(2,I,J)
738   340 CONTINUE
739   341 CONTINUE
740   342 CONTINUE
741       RETURN
742       END
743       SUBROUTINE OUTPUTS
744 C
745 C     THIS SUBROUTINE PRINTS THE RESULTS
746 C
747 C     IF STREAM FLAG IS -VE,STREAM IS NOT PRINTED
748 C     IF EQUIPMENT NUMBER IS -VE,1ST 5 EQUIPMENT PARAMETERS ARE PRINTED
749 C     F PST EQUIPMENT PARAMETER IS ALWAYS MASS OF
750 C          COMPONENT TWO
751 C
752       COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
753       COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,ND,N1ST,NFIN
754       COMMON /GERR/ JSTART,IMETH,TIME,H,HH,HINC,EPS,T,INTFL
755       COMMON /IO/ NIN,NOUT,NERR,NPOINT
756       IF (JSTART.EQ.0) WRITE (NOUT,30)
757       WRITE (NOUT,40) TIME
758       WRITE (NOUT,50)
759       DO 10 I=1,NS
760       IF (S(1,I,2).LT.0.0) GO TO 10
761       NN=S(1,I,1)+0.01
762       WRITE (NOUT,70) NN,(S(1,I,J),J=3,NC5)
763 10    CONTINUE
764       DO 20 I=1,NE
765       IF (MP(I,1).GT.0) GO TO 20
766       K=IABS(MP(I,1))
767       WRITE (NOUT,60) K,(EP(I,J),J=1,5)
768 20    CONTINUE
769       RETURN
770 C
771 C
772 30    FORMAT (1H1)
773 40    FORMAT (///1X,5(1H$),5X,30HPROCESS VARIABLES AT TIME =   ,E11.5,5X
774      1,5(1H$)/)
775 50    FORMAT (47H STREAM       FLOW      TEMP     PRES    COMPONENTS,/)
776 60    FORMAT (1H0,5X,I3,5X,5F15.5)
777 70    FORMAT (1X,I5,F12.3,2F8.1,1X,5F9.5,/,35X,5F9.5)
778       END
779       SUBROUTINE SAVEP
780 C
781 C     THIS SUBROUTINE SAVES POINTS FOR USE BY GRAPHICAL OUTPUT ROUTINES
782 C
783       COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
784       COMMON /PLT/ NPLOTS,PLOTI,PLOTD(15,4),PLOTT
785       COMMON /GERR/ JSTART,IMETH,TIME,H,HH,HINC,EPS,T,INTFL
786       COMMON /GRAPHC/ NPTS,TPLOT(1500),YPLOT(1500,15),XPLOT(1000)
787       COMMON /IO/ NIN,NOUT,NERR,NPOINT
788       COMMON/LARM/STD,PRCNT,EMC,EMSTD,EMPRCT
789 C
790 C     NOTE - SUBROUTINE IS DIMENSIONED FOR NPTMAX POINTS
791 C
792 C          -- SEE TPLOT AND YPLOT ABOVE
793 C
794       DATA NPTMAX/1500/
795 C
796 C          INITIAL POINT IS SAVED
797 C
798       IF (JSTART.EQ.0) GO TO 10
799 C
```

```
800 C           DETERMINE WHETHER TO SAVE POINT
801 C           PLOTT IS TIME VALUE OF PREVIOUS POINT SAVED
802 C           PLOTI IS APPROX TWICE INCREMENT OF TIME
803 C
804         XSPACE=(TIME-PLOTT)/PLOTI
805         NSPACE=XSPACE
806         YSPACE=NSPACE
807         IF ((XSPACE-YSPACE).GT.0.5) NSPACE=NSPACE+1
808         IF (NSPACE.LT.1) RETURN
809 10      CONTINUE
810 C
811 C           NPTS COUNTS NUMBER OF POINTS, MUST BE .LE.NPTMAX
812 C
813         NPTS=NPTS+1
814         IF (NPTS.GT.NPTMAX) GO TO 30
815         DO 20 I=1,NPLOTS
816         N1=PLOTD(I,1)+0.01
817         N2=PLOTD(I,2)+0.01
818 C
819 C           STORE DEPENDENT VALUE IN YPLOT
820 C
821         YPLOT(NPTS,I)=S(1,N1,N2)
822         DUM=YPLOT(NPTS,I)
823         CALL NOISE (1,DUM,EMSTD,EMPRCT,EMC)
824         YPLOT(NPTS,I)=DUM
825 20      CONTINUE
826 C
827 C           STORE TIME VALUE IN TPLOT
828 C
829         TPLOT(NPTS)=TIME
830 C
831 C           UPDATE PREVIOUS TIME VALUE
832 C
833         PLOTT=TIME
834         RETURN
835 30      WRITE (NERR,40)
836         CALL EXIT(1)
837 C
838 C
839 40      FORMAT (36H NOT ENOUGH POINTS IN COMMON/GRAPHC/)
840         END
841         SUBROUTINE WRITEP
842 C
843 C       THIS SUBROUTINE WRITES THE SAVED POINTS ONTO DEVICE NPOINT
844 C       THIS COULD BE A DISK FILE OR LINE PRINTER
845 C
846         COMMON /GERR/ JSTART,IMETH,TIME,H,HH,HINC,EPS
847         COMMON /GRAPHC/ NPTS,TPLOT(1500),YPLOT(1500,15),XPLOT(1000)
848         COMMON /PLT/ NPLOTS,PLOTI,PLOTD(15,4),PLOTT,PTYPE,NSTATE
849         COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
850         COMMON /IO/ NIN,NOUT,NERR,NPOINT
851         WRITE (NPOINT,100) TIME,(XPLOT(K),K=1,NSTATE)
852 20      CONTINUE
853         RETURN
854 C
855 C
856 30      FORMAT (1H1)
857 40      FORMAT (22H NUMBER OF VARIABLES =,I3,/)
858 50      FORMAT (19H NUMBER OF POINTS =,I4,///)
859 60      FORMAT (7H COLUMN,9X,8HVARIABLE,24X,5HRANGE,/)
860 70      FORMAT (4X,1H1,7X,4HTIME,18X,2E14.5)
861 80      FORMAT (2X,I3,7X,6HSTREAM,I3,1X,A4,2H (,I2,1H),3X,2E14.5)
862 90      FORMAT (/)
863 100     FORMAT (6F12.3)
864         END
865         SUBROUTINE LPLOT
866 C
867 C       THIS SUBROUTINE CREATES A LINEPLOT OF THE DESIRED OUTPUT VARIABLES
868 C       LINEPLOT CAN BE EITHER 50 OR 100 SPACES WIDE
869 C
870         COMMON /PLT/ NPLOTS,PLOTI,PLOTD(15,4),PLOTT,PTYPE
871         COMMON /GRAPHC/ NPTS,TPLOT(1500),YPLOT(1500,15),XPLOT(1000)
872         COMMON /IO/ NIN,NOUT,NERR,NPOINT
```

```
      DIMENSION ISTRM(11), ISYM(15), ILINE(110)
C
C           FIFTEEN LINES CAN BE PLOTTED
C
      DATA ISYM/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,
     1 1HO/
      DATA ISTRM/4HSTRM,4HFLAG,4HFLOW,4HTEMP,4HPRES,4HCMP1,4HCMP2,4HCMP3
     1,4HCMP4,4HCMP5,4HCMP6/
C
C           DETERMINE WIDTH OF LINEPLOT
C
      SCALE=100.0
      NTYPE=PTYPE+0.01
      IF (NTYPE.NE.0) SCALE=50.0
      NSCALE=SCALE+0.01
      WRITE (NERR,160)
      DO 10 I=1,NPLOTS
      N1=PLOTD(I,1)+0.01
      N2=PLOTD(I,2)+0.01
      WRITE (NERR,170) ISYM(I),N1,ISTRM(N2),PLOTD(I,3),PLOTD(I,4)
10    CONTINUE
      N1=NSCALE/5+1
      N2=15-NSCALE/10
      DO 20 I=1,N1
      ILINE(I)=N2*(I-1)
20    CONTINUE
      WRITE (NERR,180) (ILINE(I),I=1,N1)
      IF (NSCALE.EQ.50) GO TO 30
      WRITE (NERR,190)
      GO TO 40
30    WRITE (NERR,200)
40    CONTINUE
C
      DO 140 K=1,NPTS
      IF (K.EQ.1) GO TO 50
C
C           CALCULATE NUMBER OF LINES TO SKIP
C
      XSPACE=(TPLOT(K)-TPLOT(K-1))/PLOTI
      NSPACE=XSPACE
      YSPACE=NSPACE
      IF ((XSPACE-YSPACE).GT.0.5) NSPACE=NSPACE+1
      GO TO 60
50    NSPACE=1
60    CONTINUE
C
C           SET UP A LINE
C
      DO 70 I=2,110
      ILINE(I)=(1H )
70    CONTINUE
      ILINE(1)=1H.
      N1=NSCALE+1
      ILINE(N1)=1H.
      NSPACE=NSPACE-1
      IF (NSPACE.EQ.0) GO TO 90
      DO 80 I=1,NSPACE
      WRITE (NERR,150) (ILINE(J),J=1,N1)
80    CONTINUE
C     SET UP NEW LINE
90    DO 130 I=1,NPLOTS
C
C           SCALE DEPENDENT VARIABLE FROM 1 TO SCALE+1
C
      XN3=(YPLOT(K,I)-PLOTD(I,3))*SCALE/(PLOTD(I,4)-PLOTD(I,3))+1.0
C
C           ROUND OFF TO NEAREST INTEGER
C
      N3=XN3+0.5
      IF (N3.LT.1.OR.N3.GT.(NSCALE+1)) GO TO 130
      IF (ILINE(N3).NE.1H .AND.ILINE(N3).NE.1H.) GO TO 100
      ILINE(N3)=ISYM(I)
      GO TO 130
```

```
  100   N1=NSCALE+3
        N2=NSCALE+9
        DO 110 J=N1,N2,3
        IF (ILINE(J).EQ.1H ) GO TO 120
  110   CONTINUE
        GO TO 130
  120   ILINE(J)=ILINE(N3)
        ILINE(J+1)=ISYM(I)
  130   CONTINUE
C
C           PRINT LINE
C
        N1=NSCALE+10
        WRITE (NERR,210) TPLOT(K),(ILINE(J),J=1,N1)
  140   CONTINUE
        RETURN
C
C
  150   FORMAT (8X,101A1)
  160   FORMAT (1H1,6HSYMBOL,5X,6HSTREAM,5X,8HVARIABLE,17X,5HRANGE)
  170   FORMAT (1H0,2X,A2,8X,I3,9X,A4,5X,2F14.5)
  180   FORMAT (///4X,21I5)
  190   FORMAT (4X,4H....,21(5HI....))
  200   FORMAT (4X,4H....,11(5HI....))
  210   FORMAT (1X,E9.3,1H-,107A1)
        END
C
        SUBROUTINE CALPLT
C
C       THIS SUBROUTINE CREATES A GRAPH OF THE DESIRED OUTPUT
C       VARIABLES ON THE DD80 PLOTTER
C       THIS SUBROUTINE MAY BE MACHINE DEPENDENT
C
        COMMON/CON/NCOMP,NC5,NE,NS,TMAX,NC3
        COMMON/PLT/NPLOTS,PLOTI,PLOTD(15,4),PLOTT
        COMMON/GRAPHC/NPTS,TPLOT(1500),YPLOT(1500,15)
        DIMENSION ISTRM(11), ISYM(15)
        DATA ISYM/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,1HN,
       11H0/
        DATA ISTRM/4HSTRM,4HFLAG,4HFLOW,4HTEMP,4HPRES,4HCMP1,4HCMP2,4HCMP3
       1,4HCMP4,4HCMP5,4HCMP6/
        DIMENSION YD(1000)
        CALL KEEPB0 (4RPLTX)
        CALL FRAME
        CALL MAPX (9,0.0,TMAX,0.0,100.0,0.1,0.98,0.1,0.7)
C
C           LABEL AXES
C
        CALL SETCH (2.0,32.0,1.0,1.0,0.0)
        CALL CRTBCD (6HOUTPUT)
        CALL SETCH (45.3,2.0,1.0,1.0,0.0)
        CALL CRTBCD (4HTIME)
C
C           PLOT K=1,NPLOTS CURVES WITH LABELS
C
        DO 20 K=1,NPLOTS
        DO 10 J=1,NPTS
C
C           SCALE DEPENDENT VARIABLE FROM 0-100
C
C
      YD(J)=(YPLOT(J,K)-PLOTD(K,3))*100.0/(PLOTD(K,4)-PLOTD(K,3))
   10 CONTINUE
C
C           PLOT CURVES WITH LABES
C
C
        CALL SETPCH (1.0,1.0,100)
        KISYM=ISYM(K)
        CALL POINTC (KISYM,TPLOT,YD,NPTS)
C
C           WRITE HEADING
C
C
        AK=K
        YL=42.0-AK
        IPLOTD=PLOTD(K,1)
```

```
      MPLOT=PLOTD(K,2)
      CALL SETCH (25.0,YL,1,0,1,0,0)
      WRITE (100,40) (ISYM(K),IPLOTD,ISTRM(MPLOT),PLOTD(K,3),
     1PLOTD(K,4))
   40 FORMAT (A1,8H STREAM ,I2,3X,A4,3X,F12.5,3H  -,F12.5)
   20 CONTINUE
      CALL SETCH (6.0,30.0,1,0,2,0,0)
      CALL CRTBCD (6HLEGEND)
      RETURN
      END
      SUBROUTINE PROPS
C
C     THIS SUBROUTINE READS THE CONSTANTS FOR COMMON PHYSICAL
C          PROPERTY CORRELATIONS
C     PROPERTIES OF WATER OR AIR MAY BE SPECIFIED
C
      COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
      COMMON /ID/ NIN,NOUT,NERR,NPOINT
      COMMON /PTAB/ TREF,R
      COMMON /PROP/ MW(6),CPL(6,6),CVA(6,6),ENT(6,6),ENV(6,6),LAM(6,6),
     1VAP(6,3),WI(6,6),DNL(6,6)
      COMMON /IPROP/ NCP,NEV,NEN,NENV,ILA,IDL
      REAL MW,LAM
C
C     DEFAULT VALUES - TEMP IN C: GAS CONSTANT IN L-ATM/K/GMOLE
C
      DATA TREF,R/0.0,0.08205/
C
      CALL GET (NAME,M,X,1)
      NPP=X+SIGN(0.01,X)
      IF (NAME.NE.4HPROP) GO TO 290
      IF (NPP.EQ.-1) GO TO 220
      IF (NPP.EQ.-2) GO TO 250
      WRITE (NOUT,320) NPP
C
      DO 210 NTIME=1,NPP
      CALL GET (NAME,M,X,1)
      NCO=X+0.01
      IF (NAME.EQ.4HMOWE) GO TO 10
      IF (NAME.EQ.4HCPLI) GO TO 30
      IF (NAME.EQ.4HCPVA) GO TO 50
      IF (NAME.EQ.4HENTL) GO TO 70
      IF (NAME.EQ.4HENTV) GO TO 90
      IF (NAME.EQ.4HLAMB) GO TO 110
      IF (NAME.EQ.4HVAPR) GO TO 130
      IF (NAME.EQ.4HWILS) GO TO 150
      IF (NAME.EQ.4HDENL) GO TO 170
      IF (NAME.EQ.4HTREF) GO TO 190
      IF (NAME.EQ.4HR    ) GO TO 200
      WRITE (NERR,300)
      STOP
   10 CONTINUE
C
C          MOLECULAR WEIGHT
C
      DO 20 I=1,NCOMP
      READ (NIN,370) MW(I)
      WRITE (NOUT,360) MW(I)
   20 CONTINUE
      GO TO 210
   30 CONTINUE
C
C          LIQUID HEAT CAPACITY
C
      NCP=NCO
      DO 40 I=1,NCOMP
      READ (NIN,370) (CPL(I,J),J=1,NCO)
      WRITE (NOUT,360) (CPL(I,J),J=1,NCO)
   40 CONTINUE
      GO TO 210
```

```
1090 50    CONTINUE
1091 C
1092 C          VAPOUR HEAT CAPACITY
1093 C
1094       NCV=NCO
1095       DO 60 I=1,NCOMP
1096       READ (NIN,370) (CVA(I,J),J=1,NCO)
1097       WRITE (NOUT,360) (CVA(I,J),J=1,NCO)
1098 60    CONTINUE
1099       GO TO 210
1100 70    CONTINUE
1101 C
1102 C          LIQUID ENTHALPY
1103 C
1104       NEN=NCO
1105       DO 80 I=1,NCOMP
1106       READ (NIN,370) (ENT(I,J),J=1,NCO)
1107       WRITE (NOUT,360) (ENT(I,J),J=1,NCO)
1108 80    CONTINUE
1109       GO TO 210
1110 90    CONTINUE
1111 C
1112 C          VAPOUR ENTHALPY
1113 C
1114       NENV=NCO
1115       DO 100 I=1,NCOMP
1116       READ (NIN,370) (ENV(I,J),J=1,NCO)
1117       WRITE (NOUT,360) (ENV(I,J),J=1,NCO)
1118 100   CONTINUE
1119       GO TO 210
1120 110   CONTINUE
1121 C
1122 C          HEAT OF VAPOURIZATION
1123 C
1124       ILA=NCO
1125       DO 120 I=1,NCOMP
1126       READ (NIN,370) (LAM(I,J),J=1,NCO)
1127       WRITE (NOUT,360) (LAM(I,J),J=1,NCO)
1128 120   CONTINUE
1129       GO TO 210
1130 130   CONTINUE
1131 C
1132 C          VAPOUR PRESSURE
1133 C
1134       DO 140 I=1,NCOMP
1135       READ (NIN,370) (VAP(I,J),J=1,3)
1136       WRITE (NOUT,360) (VAP(I,J),J=1,3)
1137 140   CONTINUE
1138       GO TO 210
1139 150   CONTINUE
1140 C
1141 C          WILSON LIQUID ACTIVITY COEFFICIENTS
1142 C
1143       DO 160 I=1,NCOMP
1144       READ (NIN,370) (WI(I,J),J=1,NLCMP)
1145       WRITE (NOUT,360) (WI(I,J),J=1,NCOMP)
1146 160   CONTINUE
1147       GO TO 210
1148 170   CONTINUE
1149 C
1150 C          LIQUID DENSITY
1151 C
1152       IDL=NCO
1153       DO 180 I=1,NCOMP
1154       READ (NIN,370) (DNL(I,J),J=1,NCO)
1155       WRITE (NOUT,360) (DNL(I,J),J=1,NCO)
1156 180   CONTINUE
1157       GO TO 210
1158 190   CONTINUE
1159 C
1160 C          REFERENCE TEMPERATURE
1161 C
1162       TREF=X
1163       GO TO 210
```

```
1164 200   CONTINUE
1165 C
1166 C           IDEAL GAS CONSTANT
1167 C
1168       R=X
1169 210   CONTINUE
1170       GO TO 240
1171 C
1172 220   CONTINUE
1173 C
1174 C         WATER(NPP=-1)
1175 C
1176       WRITE (NOUT,340)
1177       TREF=0.
1178       DO 230 I=1,NCOMP
1179       MW(I)=18.
1180       CPL(I,1)=18.
1181       ENT(I,1)=0.
1182       ENT(I,2)=18.
1183       DNL(I,1)=62.4/18.
1184 230   CONTINUE
1185 240   CALL GET (NAME,M,X,1)
1186       IF (NAME.NE.4HEND ) GO TO 200
1187       RETURN
1188 250   CONTINUE
1189 C
1190 C         AIR(NPP=-2)
1191 C
1192       WRITE (NOUT,350)
1193       TREF=0.
1194       DO 260 I=1,NCOMP
1195       MW(I)=29.
1196 260   CONTINUE
1197       DO 270 NTIME=1,2
1198       CALL GET (NAME,M,X,1)
1199       IF (NAME.EQ.4HEND ) RETURN
1200       IF (NAME.EQ.4HR   ) R=X
1201 270   CONTINUE
1202 C
1203 280   WRITE (NERR,310)
1204       STOP
1205 290   WRITE (NERR,330)
1206       STOP
1207 C
1208 C
1209 C     FREE FORMAT (DEC SYSTEM-10)
1210 C
1211 C  37 FORMAT(5F)
1212 C
1213 300   FORMAT (25H ERROR IN PROPERTIES DATA)
1214 310   FORMAT (23H END CARD SHOULD FOLLOW)
1215 320   FORMAT (15H PROPERTY TABLE,I4,8H ENTRIES)
1216 330   FORMAT (30H PROPERTIES CARD SHOULD FOLLOW)
1217 340   FORMAT (20H PROPERTIES AS WATER)
1218 350   FORMAT (18H PROPERTIES AS AIR)
1219 360   FORMAT (12X,5F12.5)
1220 370   FORMAT (12X,5F12.5)
1221       END
1222 C
1223       SUBROUTINE MOWE (IU,IS,MOW)
1224 C
1225 C       IU = 1
1226 C       IS-STREAM NUMBER
1227 C       MOW-AVERAGE MOLECULAR WEIGHT
1228 C
1229       COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
1230       COMMON /CON/ NCOMP,NCS,NE,NS,TMAX,NC3,N8,N1,N2,NF
1231       COMMON /PROP/ MW(6),CPL(6,6),CVA(6,6),ENT(6,6),ENV(6,6),LAM(6,6),
1232      1VAP(6,3),WI(6,6),DNL(6,6)
1233       REAL MW,MOW
1234 C
1235       MOW=0.
1236       DO 10 I=1,NCOMP
1237       NC=I+5
```

```
      MOW=MOW+MW(I)*S(IU,IS,NC)
   10 CONTINUE
      RETURN
      END
      SUBROUTINE CPLI (IU,IS,C)
C
C       IU = 1
C       IS-LIQUID STREAM NUMBER
C       C-AVERAGE LIQUID HEAT CAPACITY
C
      COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
      COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
      COMMON /PTAB/ TREF,R
      COMMON /PROP/ MW(6),CPL(6,6),CVA(6,6),ENT(6,6),ENV(6,6),LAM(6,6),
     1VAP(6,3),WI(6,6),DNL(6,6)
      COMMON /IPROP/ NCP,NCV,NEN,NENV,ILA,IDL
C
      T=S(IU,IS,4)+TREF
      C=0.
      DO 20 I=1,NCOMP
      NC=I+5
      CP=CPL(I,1)
      DO 10 J=2,NCP
      CP=CP+CPL(I,J)*T**(J-1)
   10 CONTINUE
      C=C+CP*S(IU,IS,NC)
   20 CONTINUE
      RETURN
      END
      SUBROUTINE CPVA (IU,IS,C)
C
C       IU = 1
C       IS-VAPOUR STREAM NUMBER
C       C-AVERAGE VAPOUR HEAT CAPACITY
C
      COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
      COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
      COMMON /PTAB/ TREF,R
      COMMON /PROP/ MW(6),CPL(6,6),CVA(6,6),ENT(6,6),ENV(6,6),LAM(6,6),
     1VAP(6,3),WI(6,6),DNL(6,6)
      COMMON /IPROP/ NCP,NCV,NEN,NENV,ILA,IDL
C
      T=S(IU,IS,4)+TREF
      C=0.
      DO 20 I=1,NCOMP
      NC=I+5
      CPV=CVA(I,1)
      DO 10 J=2,NCV
      CPV=CPV+CVA(I,J)*T**(J-1)
   10 CONTINUE
      C=C+S(IU,IS,NC)*CPV
   20 CONTINUE
      RETURN
      END
      SUBROUTINE ENTL (IU,IS,E,DE)
C
C       IU = 1
C       IS-LIQUID STREAM NUMBER
C       E-AVERAGE LIQUID ENTHALPY
C       DE-AVERAGE DERIVATIVE OF LIQUID ENTHALPY
C       (WITH RESPECT TO TEMPERATURE)
C
      COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
      COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
      COMMON /PTAB/ TREF,R
      COMMON /PROP/ MW(6),CPL(6,6),CVA(6,6),ENT(6,6),ENV(6,6),LAM(6,6),
     1VAP(6,3),WI(6,6),DNL(6,6)
      COMMON /IPROP/ NCP,NCV,NEN,NENV,ILA,IDL
C
      T=S(IU,IS,4)+TREF
      E=0.
      DE=0.
      DO 20 I=1,NCOMP
```

```
      NC=I+5
      EN=ENT(I,1)
      DEN=ENT(I,2)
      DO 10 J=2,NEN
      EN=EN+ENT(I,J)*T**(J-1)
      IF (J.GE.3) DEN=DEN+(J-1)*ENT(I,J)*T**(J-2)
10    CONTINUE
      E=E+S(IU,IS,NC)*EN
      DE=DE+S(IU,IS,NC)*DEN
20    CONTINUE
      RETURN
      END

      SUBROUTINE ENTV (IU,IS,E,DE)
C
C       IU = 1
C       IS-VAPOUR STREAM NUMBER
C       E-AVERAGE VAPOUR ENTHALPY
C       DE-AVERAGE DERIVATIVE OF THE VAPOUR ENTHALPY
C       (WITH RESPECT TO TEMPERATURE)
C
      COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
      COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
      COMMON /PTAB/ TREF,R
      COMMON /PROP/ MW(6),CPL(6,6),CVA(6,6),ENT(6,6),ENV(6,6),LAM(6,6),
     1VAP(6,3),WI(6,6),DNL(6,6)
      COMMON /IPROP/ NCP,NCV,NEN,NENV,ILA,IDL
C
      T=S(IU,IS,4)+TREF
      E=0.
      DO 20 I=1,NCOMP
      NC=I+5
      EN=ENV(I,1)
      DEN=ENV(I,2)
      DO 10 J=2,NEN
      EN=EN+ENV(I,J)*T**(J-1)
      IF (J.GE.3) DEN=DEN+(J-1)*ENV(I,J)*T**(J-2)
10    CONTINUE
      E=E+S(IU,IS,NC)*EN
      DE=DE+S(IU,IS,NC)*DEN
20    CONTINUE
      RETURN
      END
      SUBROUTINE LAMB (IU,IS,L)
C
C       IU = 1
C       IS-VAPOUR STREAM NUMBER
C       L-HEAT OF VAPOURIZATION
C
      COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
      COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
      COMMON /PTAB/ TREF,R
      COMMON /PROP/ MW(6),CPL(6,6),CVA(6,6),ENT(6,6),ENV(6,6),LAM(6,6),
     1VAP(6,3),WI(6,6),DNL(6,6)
      COMMON /IPROP/ NCP,NCV,NEN,NENV,ILA,IDL
      REAL LAM,LA,L
C
      T=S(IU,IS,4)+TREF
      L=0.
      DO 20 I=1,NCOMP
      NC=I+5
      LA=LAM(I,1)
      DO 10 J=2,ILA
      LA=LA+LAM(I,J)*T**(J-1)
10    CONTINUE
      L=L+S(IU,IS,NC)*LA
20    CONTINUE
      RETURN
      END
      SUBROUTINE DENL (IU,IS,D)
C
C       IU = 1
C       IS-LIQUID STREAM NUMBER
C       D-AVERAGE LIQUID MOLAL DENSITY
```

```
1384 C
1385       COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
1386       COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
1387       COMMON /PTAB/ TREF,R
1388       COMMON /PROP/ MW(6),CPL(6,6),CVA(6,6),ENT(6,6),ENV(6,6),LAM(6,6),
1389      1VAP(6,3),WI(6,6),DNL(6,6)
1390       COMMON /IPROP/ NCP,NCV,NEN,NENV,ILA,IDL
1391 C
1392       T=S(IU,IS,4)+TREF
1393       D=0.0
1394       DO 20 I=1,NCOMP
1395       NC=I+5
1396       DL=DNL(I,1)
1397       DO 10 J=2,IDL
1398       DL=DL+DNL(I,J)*T**(J-1)
1399 10    CONTINUE
1400       D=D+S(IU,IS,NC)*DL
1401 20    CONTINUE
1402       RETURN
1403       END
1404       SUBROUTINE DENV (IU,IS,D)
1405 C
1406 C       IU = 1
1407 C       IS-VAPOUR STREAM NUMBER
1408 C       D-AVERAGE VAPOUR MOLAL DENSITY
1409 C       ASSUME VAPOUR BEHAVES AS IDEAL GAS
1410 C
1411 C       CALLS TO SUBROUTINE MOWE(IU,IS,M)
1412 C       M-AVERAGE MOLAL MOLECULAR WEIGHT
1413 C
1414       COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
1415       COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
1416       COMMON /PTAB/ TREF,R
1417       REAL M
1418 C
1419       T=S(IU,IS,4)+TREF
1420       P=S(IU,IS,5)
1421       CALL MOWE (IU,IS,M)
1422       D=P/R/T*M
1423       RETURN
1424       END
1425       SUBROUTINE VAPR (IU,IS,PV,DPV)
1426 C
1427 C       IU = 1
1428 C       IS-LIQUID STREAM NUMBER
1429 C       PV-VAPOUR PRESSURE FOR EACH COMPONENT
1430 C       DPV-DERIVATIVE OF VAPOUR PRESSURE FOR EACH COMPONENT
1431 C       (WITH RESPECT TO TEMPERATURE)
1432 C
1433       COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
1434       COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
1435       COMMON /PTAB/ TREF,R
1436       COMMON /PROP/ MW(6),CPL(6,6),CVA(6,6),ENT(6,6),ENV(6,6),LAM(6,6),
1437      1VAP(6,3),WI(6,6),DNL(6,6)
1438       DIMENSION PV(6), DPV(6)
1439 C
1440       T=S(IU,IS,4)+TREF
1441       DO 10 I=1,NCOMP
1442       PV(I)=EXP(VAP(I,1)+VAP(I,2)/(VAP(I,3)+T))
1443       DPV(I)=-PV(I)*VAP(I,2)/(VAP(I,3)+T)**2
1444 10    CONTINUE
1445       RETURN
1446       END
1447       SUBROUTINE WILS (IU,IS,A)
1448 C
1449 C       IU = 1
1450 C       IS-LIQUID STREAM NUMBER
1451 C       A-WILSON LIQUID ACTIVITY COEFFICIENTS FOR EACH COMPONENT
1452 C
1453       COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
1454       COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
1455       COMMON /PTAB/ TREF,R
1456       COMMON /PROP/ MW(6),CPL(6,6),CVA(6,6),ENT(6,6),ENV(6,6),LAM(6,6),
```

```
1457       1VAP(6,3),WI(6,6),DNL(6,6)
1458       DIMENSION SI(6), A(6)
1459 C
1460       IF (WI(1,1).LT.1.E-05) GO TO 50
1461       DO 20 I=1,NCOMP
1462       SI(I)=0.
1463       DO 10 J=1,NCOMP
1464       NCJ=J+5
1465       SI(I)=SI(I)+WI(I,J)*S(IV,IS,NCJ)
1466 10    CONTINUE
1467 20    CONTINUE
1468       DO 40 K=1,NCOMP
1469       SUM=0.
1470       DO 30 I=1,NCOMP
1471       NCI=I+5
1472       SUM=SUM+WI(I,K)*S(IV,IS,NCI)/SI(I)
1473 30    CONTINUE
1474       A(K)=EXP(1.-ALOG(SI(K))-SUM)
1475 40    CONTINUE
1476       RETURN
1477 50    CONTINUE
1478       DO 60 I=1,NCOMP
1479       A(I)=1.
1480 60    CONTINUE
1481       RETURN
1482       END
1483       SUBROUTINE BUBL (IL,IV)
1484 C
1485 C        IL-LIQUID STREAM NUMBER
1486 C        IV-VAPOUR STREAM NUMBER
1487 C
1488 C        CALCULATE BUBBLE POINT TEMPERATURE AND PLACE IN STREAM
1489 C        ASSUME VAPOUR AND LIQUID STREAMS IN EQUILIBRIUM
1490 C
1491 C        CALLS TO SUBROUTINE WILS(IG,IL,A)
1492 C        A-WILSON LIQUID ACTIVITY COEFFICIENTS FOR EACH COMPONENT
1493 C
1494 C        CALLS TO SUBROUTINE VAPR(IG,IL,PV,DPV)
1495 C        PV-VAPOUR PRESSURE
1496 C        DPV-DERIVATIVE OF VAPOUR PRESSURE WITH RESPECT TO TEMPERATURE
1497 C
1498 C        NEWTON-RAPHSON ITERATIVE METHOD IS USED
1499 C
1500       COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
1501       COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,N8,N1,N2,NF
1502       COMMON /PTAB/ TREF,R
1503       COMMON /IO/ NIN,NOUT,NERR,NPOINT
1504       DIMENSION A(6), PV(6), DPV(6)
1505 C
1506       IG=1
1507       T=S(IG,IL,4)+TREF
1508       P=S(IG,IL,5)
1509       CALL WILS (IG,IL,A)
1510       K=0
1511 10    CONTINUE
1512       IF (K.GE.20) GO TO 30
1513       K=K+1
1514       CALL VAPR (IG,IL,PV,DPV)
1515       SY=0.
1516       SDY=0.
1517       DO 20 I=1,NCOMP
1518       NC=I+5
1519       S(IG,IV,NC)=S(IG,IL,NC)*PV(I)*A(I)/P
1520       SY=SY+S(IG,IV,NC)
1521       DY=-S(IG,IL,NC)*DPV(I)*A(I)/P
1522       SDY=SDY+DY
1523 20    CONTINUE
1524       T=T-(1.-SY)/SDY
1525       S(IG,IL,4)=T-TREF
1526       IF (ABS(1.-SY).GT.0.01) GO TO 10
1527       S(IG,IV,4)=S(IG,IL,4)
1528       RETURN
1529 C
```

```
1530 30     CONTINUE
1531        WRITE (NERR,40) K
1532        STOP
1533 C
1534 C
1535 40     FORMAT (38H SUBROUTINE BUBL, ITERATIONS MORE THAN,I3)
1536        END
1537        SUBROUTINE KVAL (IU,IS,HKK)
1538 C
1539 C         IU = 1
1540 C         IS-LIQUID STREAM NUMBER
1541 C         HKK-VAPOUR-LIQUID EQUILIBRIUM RATIO FOR EACH COMPONENT
1542 C
1543 C         CALLS TO SUBROUTINE VAPR(IU,IS,PV,DPV)
1544 C         PV-VAPOUR PRESSURE
1545 C         DPV-DERIVATIVE OF VAPOUR PRESSURE WITH RESPECT TO TEMPERATURE
1546 C
1547 C         CALLS TO WILS(IU,IL,A)
1548 C         A-WILSON LIQUID ACTIVITY COEFFICIENTS FOR EACH COMPONENT
1549 C
1550        COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
1551        COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,N8,N1,N2,NF
1552        DIMENSION HKK(6), A(6), PV(6), DPV(6)
1553 C
1554        P=S(IU,IS,5)
1555        CALL WILS (IU,IS,A)
1556        CALL VAPR (IU,IS,PV,DPV)
1557        DO 10 I=1,NCOMP
1558        HKK(I)=PV(I)*A(I)/P
1559 10     CONTINUE
1560        RETURN
1561        END
1562 C
1563 C
1564        SUBROUTINE TEMPL (Q,IL)
1565 C
1566 C         Q-MOLAL HEAT CONTENT
1567 C         IL-LIQUID STREAM NUMBER
1568 C
1569 C         CALCULATE EXIT TEMPERATURE OF MIXED LIQUID STREAMS
1570 C         WHERE ENTHALPY IS FUNCTION OF TEMPERATURE
1571 C         PLACE TEMPERATURE IN STREAM
1572 C
1573 C         CALLS TO ENTL(IG,IL,E,DE)
1574 C         E-AVERAGE LIQUID ENTHALPY
1575 C         DE-AVERAGE DERIVATIVE OF LIQUID ENTHALPY
1576 C         (WITH RESPECT TO TEMPERATURE)
1577 C
1578        COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
1579        COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,N8,N1,N2,NF
1580        COMMON /PTAB/ TREF,R
1581        COMMON /IO/ NIN,NOUT,NERR,NPOINT
1582 C
1583        IG=1
1584        T=S(IG,IL,4)+TREF
1585        T1=T
1586        K=0
1587 10     CONTINUE
1588        T=T1
1589        IF (K.GE.20) GO TO 20
1590        K=K+1
1591        CALL ENTL (IG,IL,E,DE)
1592        F=Q-E
1593        DF=-DE
1594        T1=T-F/DF
1595        S(IG,IL,4)=T-TREF
1596        IF (ABS(T-T1).GT.0.01) GO TO 10
1597        RETURN
1598 C
1599 20     CONTINUE
1600        WRITE (NERR,30) K
1601        STOP
1602 C
```

```
1603 C
1604 30    FORMAT (39H SUBROUTINE TEMPL, ITERATION MORE THAN,I3)
1605       END
1606       SUBROUTINE TEMPV (Q,IV)
1607 C
1608 C       Q-MOLAL HEAT CONTENT
1609 C       IV-STREAM NUMBER
1610 C
1611 C       CALCULATE EXIT TEMPERATURE OF MIXED VAPOUR STREAMS
1612 C       WHERE ENTHALPY IS FUNCTION OF TEMPERATURE
1613 C       PLACE TEMPERATURE IN STREAM
1614 C
1615 C       CALLS TO ENTV(IG,IV,E,DE)
1616 C       E-AVERAGE VAPOUR ENTHALPY
1617 C       DE-AVERAGE DERIVATIVE OF VAPOUR ENTHALPY
1618 C       (WITH RESPECT TO TEMPERATURE)
1619 C
1620       COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
1621       COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1,N2,NF
1622       COMMON /PTAB/ TREF,R
1623       COMMON /IO/ NIN,NOUT,NERR,NPOINT
1624 C
1625       IG=1
1626       T=S(IG,IV,4)+TREF
1627       T1=T
1628       K=0
1629 10    CONTINUE
1630       T=T1
1631       IF (K.GE.20) GO TO 20
1632       K=K+1
1633       CALL ENTV (IG,IV,E,DE)
1634       F=Q-E
1635       DF=-DE
1636       T1=T-F/DF
1637       S(IG,IV,4)=T-TREF
1638       IF (ABS(T1-T).GT.0.01) GO TO 10
1639       RETURN
1640 C
1641 20    CONTINUE
1642       WRITE (NERR,30) K
1643       STOP
1644 C
1645 C
1646 30    FORMAT (39H SUBROUTINE TEMPV, ITERATIONS MORE THAN,I3)
1647       END
1648       SUBROUTINE TYPE1
1649 C
1650 C     SUBROUTINE STGIDL
1651 C
1652 C                    ONE IDEAL STAGE FOR EXTRACTION WITH SPECIFIED
1653 C           PHASE VOLUMES - SPECIFIC TO CO-EXTRACTION OF U AND PU
1654 C           FROM OTHER SALTS - EACH STAGE MAY HAVE ANY NUMBER OF FEEDS
1655 C           AND EFFLUENTS UP TO FOUR EACH - CODE MAY BE GENERALIZED
1656 C           TO GENERAL EXTRACTION OF ANY SYSTEM AND TO VARIABLE PHASE
1657 C           VOLUMES - EXTENSION TO PU+3 - U+6 SEPARATION POSSIBLE
1658 C
1659       COMMON/MAT/MP(35,13),EP(35,10),S(2,45,13),EX(50)
1660       COMMON/CON/NCOMP,NC5,NE,NS,TMAX,NC3,NB,N1ST,NFIN
1661       COMMON/GERR/JSTART,IMETH,TIME,H,HINC,EPS,TT,INTFL
1662       COMMON/UNIT/IM,NMP
1663       COMMON/IO/NIN,NOUT,NERR,NPOINT
1664       REAL LIN,LOUT,KV,KEXT
1665       INTEGER OUTPRO,OUTSTO,OUTPRA,OUTSTA
1666       DIMENSION YO(10),XO(10),XSLF(10),XD(13,8),TD(8)
1667       DIMENSION X(13,8),W(8),T(8),XIN(10),VOL(2),XOUT(10),DERY(10)
1668       DIMENSION YIN(10),YOUT(10),DISCO(10),YIDEAL(10),P(8),DERX(10)
1669       DIMENSION XX(13,8),XXX(13,8),TRANS(10)
1670 C
1671 C
1672 C
1673 C        EQUIPMENT PARAMETERS
1674 C
1675 C            1 - PU HOLDUP, G
```

```
1676 C            2 - VOL(NO)=ORGANIC PHASE VOLUME
1677 C            3 - VOL(NA)=AQUEOUS PHASE VOLUME
1678 C            4 - KV=RATE CONSTANT FOR VOLUME CHANGE
1679 C            5 - KEXT=RATE CONSTANT FOR APPROACH TO EQUIL. EXTRACTION
1680 C            6 - XTBP=VOLUME FRACTION TRIBUTYL PHOSPHATE IN ORGANIC PH.
1681 C
1682 C         COMPONENT NUMBERS
1683 C
1684 C            1 - U
1685 C            2 - PU
1686 C            3 - HNO3
1687 C            4 - NO3- SALT
1688 C         CONSTANT PARAMETERS
1689 C
1690       VOL(1)=EP(IM,2)
1691       VOL(2)=EP(IM,3)
1692       VOLT=VOL(1)+VOL(2)
1693       XTBP=EP(IM,6)
1694       KV=EP(IM,4)
1695       KEXT=EP(IM,5)
1696 C
1697       IF (INTFL.EQ.0) GO TO 2
1698       RETURN
1699     2 CONTINUE
1700 C        IDENTIFY IN AND OUT STREAMS
1701 C
1702       INSTGO=IABS(MP(IM,3))
1703       INFDO=IABS(MP(IM,4))
1704       OUTPRO=IABS(MP(IM,5))
1705       OUTSTO=IABS(MP(IM,6))
1706       INSTGA=IABS(MP(IM,7))
1707       INFDA=IABS(MP(IM,8))
1708       OUTPRA=IABS(MP(IM,9))
1709       OUTSTA=IABS(MP(IM,10))
1710 C
1711 C     NUMBER OF ODE'S
1712 C
1713       NEQ=NCOMP+1
1714 C
1715 C     INITIAL VALUES OF W(J),T(J),P(J),AND X(I,J)
1716 C
1717       NST=8
1718       DO 26 I=3,NC5
1719       DO 25 J=1,NST
1720       IA=IABS(MP(IM,J+2))
1721       IF(IA.EQ.0)  GO TO 24
1722       XX(I-2,J)=S(1,IA,I)
1723       GO TO 25
1724    24 XX(I-2,J)=0.0
1725    25 CONTINUE
1726    26 CONTINUE
1727 C
1728 C        RENAME STREAM VARIABLES AND CONVERT TO MOLAR CONCENTRATIONS
1729 C
1730       DO 30 J=1,NST
1731       W(J)=XX(1,J)
1732       T(J)=XX(2,J)
1733       P(J)=XX(3,J)
1734       DO 29 I=4,NC3
1735       X(I-3,J)=XX(I,J)
1736    29 CONTINUE
1737       X(1,J)=X(1,J)/238.0
1738       X(2,J)=X(2,J)/239.0
1739    30 CONTINUE
1740 C
1741 C        INPUT AND OUTPUT PHASE COMPOSITIONS AND FLOWRATES ASSUMING
1742 C        CONSTANT DENSITY; TEMP. ASSUMING NO HEAT TRANSFER OR GENERATION
1743 C
1744       LIN=0.0
1745       LOUT=0.0
1746       IF (VOL(2).EQ.0.0) GO TO 41
1747       LIN=W(5)+W(6)
1748       W(7)=W(7)*(W(5)+W(6))/(W(7)+W(8))
```

```
1749        W(8)=W(8)*(W(5)+W(6))/(W(7)+W(8))
1750        LOUT=W(7)+W(8)
1751     41 CONTINUE
1752        VIN=W(1)+W(2)
1753        W(3)=W(3)*(W(1)+W(2))/(W(3)+W(4))
1754        W(4)=W(4)*(W(1)+W(2))/(W(3)+W(4))
1755        VOUT=W(3)+W(4)
1756        DO 40 I=1,NCOMP
1757        IF (VOL(2).EQ.0.0) GO TO 43
1758        XIN(I)=(X(I,5)*W(5)+X(I,6)*W(6))/LIN
1759        XOUT(I)=(X(I,7)*W(7)+X(I,8)*W(8))/LOUT
1760     43 YIN(I)=(X(I,1)*W(1)+X(I,2)*W(2))/VIN
1761        YOUT(I)=(X(I,3)*W(3)+X(I,4)*W(4))/VOUT
1762        YOUT(I+NEQ)=XOUT(I)
1763     40 CONTINUE
1764        TIN=(T(1)*W(1)+T(2)*W(2)+T(5)*W(5)+T(6)*W(6))/(LIN+VIN)
1765        TOUT=(T(3)*W(3)+T(4)*W(4)+T(7)*W(7)+T(8)*W(8))/(LOUT+VOUT)
1766 C
1767 C         BEGINNING OF DERIVATIVE COMP. FOR SOLVING ODE'S AND CONVERGING
1768 C         MATERIAL BALANCE
1769 C
1770        YOUT(NEQ)=TOUT
1771 C    1 CONTINUE
1772        IF(VOL(2).EQ.0.0) GO TO 61
1773 C
1774 C       CONVERT TO SOLUTE FREE VARIABLES
1775 C
1776     7 CONTINUE
1777        CALL CONVT1(X,T,XTBP,DENOMA,DENOMO)
1778 C
1779 C       IDEAL ORGANIC PHASE EQUIL. COMPOSITION
1780 C
1781        DO 42 I=1,NCOMP
1782        XSLF(I)=X(I,8)
1783     42 CONTINUE
1784        CALL ORGPH(TOUT,XTBP,XSLF,DISCO)
1785        DO 50 I=1,NCOMP
1786        YIDEAL(I)=XOUT(I)*DISCO(I)*DENOMO/DENOMA
1787 C
1788 C         DERIVATIVES FOR ODE'S - THE EQUATIONS USED DO NOT ALLOW FOR
1789 C         PHASE VOLUME CHANGE
1790 C
1791        TRANS(I)=YIDEAL(I)-YOUT(I)
1792        DERY(I)=(VIN*YIN(I)-VOUT*YOUT(I)+KEXT*(TRANS(I)))/VOL(1)
1793     50 CONTINUE
1794        DERY(NEQ)=((VIN+LIN)*TIN-(VOUT+LOUT)*TOUT)/(VOL(1)+VOL(2))
1795        DO 60 I=1,NCOMP
1796        DERX(I)=(LIN*XIN(I)-LOUT*XOUT(I)-KEXT*(TRANS(I)))/VOL(2)
1797     60 CONTINUE
1798     61 IF(VOL(2).NE.0.0)  GO TO 63
1799 C
1800 C              STIRRED TANK SECTION
1801 C
1802        DO 62 I=1,NCOMP
1803        DERY(I)=(VIN*YIN(I)-VOUT*YOUT(I))/VOL(1)
1804        DERX(I)=0.
1805     62 CONTINUE
1806        DERY(NEQ)=(VIN*TIN-VOUT*TOUT)/VOL(1)
1807     63 CONTINUE
1808        TOUT=YOUT(NEQ)
1809 C
1810 C         CALCULATE RETURN VALUES
1811 C
1812        DO 70 I=1,NCOMP
1813        XD(I,7)=DERX(I)
1814        XD(I,8)=XD(I,7)
1815        XD(I,3)=DERY(I)
1816        XD(I,4)=XD(I,3)
1817     70 CONTINUE
1818        TD(7)=DERY(NEQ)
1819        TD(8)=TD(7)
1820        TD(3)=TD(7)
1821        TD(4)=TD(7)
1822 C
```

```
C      RENAME STREAM VARIABLES AND CONVERT TO MASS CONCENTRATIONS OF U-PU
C
       DO 91 J=3,4
       XXX(7,J)=XD(4,J)
       XXX(6,J)=XD(3,J)
       XXX(5,J)=XD(2,J)*239.0
       XXX(4,J)=XD(1,J)*238.0
       XXX(3,J)=0.0
       XXX(2,J)=TD(J)
       XXX(1,J)=0.0
    91 CONTINUE
       DO 92 J=7,8
       XXX(7,J)=XD(4,J)
       XXX(6,J)=XD(3,J)
       XXX(5,J)=XD(2,J)*239.0
       XXX(4,J)=XD(1,J)*238.0
       XXX(3,J)=0.0
       XXX(2,J)=TD(J)
       XXX(1,J)=0.0
    92 CONTINUE
C
C      CALCULATE OUTPUT TO STREAM VARIABLE FILE
C
       IF (OUTPRO.EQ.0) GO TO 100
       S(1,OUTPRO,3)=W(3)
   100 IF (OUTSTO.EQ.0) GO TO 101
       S(1,OUTSTO,3)=W(4)
   101 IF (OUTPRA.EQ.0) GO TO 102
       S(1,OUTPRA,3)=W(7)
   102 IF (OUTSTA.EQ.0) GO TO 103
       S(1,OUTSTA,3)=W(8)
   103 CONTINUE
       DO 88 I=1,NC3
       IF(OUTPRO.EQ.0) GO TO 83
       S(2,OUTPRO,I+2)=XXX(I,3)
    83 IF(OUTSTO.EQ.0) GO TO 86
       S(2,OUTSTO,I+2)=XXX(I,4)
    86 IF(OUTPRA.EQ.0) GO TO 87
       S(2,OUTPRA,I+2)=XXX(I,7)
    87 IF(OUTSTA.EQ.0) GO TO 88
       S(2,OUTSTA,I+2)=XXX(I,8)
    88 CONTINUE
       EP(IM,1)=S(1,OUTSTO,7)*VOL(1)+S(1,OUTSTA,7)*VOL(2)
       RETURN
       END
       SUBROUTINE CONVTL(X,T,XTBP,DENOMA,DENOMO)
       COMMON/UNIT/IM,NMP
       COMMON/CON/NCOMP,NCS,NE,NS,TMAX,NC3
       DIMENSION X(13,8),T(8),W(8)
C
C         CONVERTS CONCENTRATIONS AND FLOW RATES TO SOLUTE FREE BASIS
C         SPECIFIC TO U-PU-HNO3-NO3-SALT SYSTEM
C
       NST=8
       TO=3.65*XTBP
       US=0.5*TO/(1.0+0.046*TO)
       PUS=0.5*TO/(1.0+0.09*TO)
C           AQUEOUS PHASE
       DO 10 J=7,NST
       DENOMA=1.0-0.0724*X(1,J)-0.13*X(2,J)-0.0309*X(3,J)-0.031*X(4,J)
       DO 9 I=1,NCOMP
       X(I,J)=X(I,J)/DENOMA
     9 CONTINUE
       W(J)=W(J)*DENOMA
    10 CONTINUE
C           ORGANIC PHASE
       DO 20 J=3,4
       WO=(3.95-0.0144*T(J))*XTBP**1.65
       HS=TO*(1.0-0.00609*WO)/(1.0+0.043*TO)
       WC=WO*(1.0-X(1,J)/US-X(2,J)/PUS-0.65*X(3,J)/HS)
       DENOMO=1.0-0.097*X(1,J)-0.139*X(2,J)-0.043*X(3,J)-0.0174*WC
       DO 19 I=1,NCOMP
       X(I,J)=X(I,J)/DENOMO
```

```
      19 CONTINUE
         W(J)=W(J)*DENOMO
      20 CONTINUE
         RETURN
         END
         SUBROUTINE CONVT2(X,T,XTBP,W)
         COMMON/UNIT/IM,NMP
         COMMON/CON/NCOMP,NC5,NE,NS,TMAX,NC3
         DIMENSION X(13,8),T(8),W(8)
C
C           CONVERTS CONCENTRATIONS AND FLOW RATES FROM SOLUTE FREE TO
C           MOLAR CONCENTRATION BASIS - SPECIFIC TO U-PU-HNO3-NO3-SALT SYS
C
         NST=8
         TO=3.65*XTBP
C              AQUEOUS PHASE
         DO 10 J=5,NST
         DENA=1.0+0.0724*X(1,J)+0.13*X(2,J)+0.0309*X(3,J)+0.031*X(4,J)
         DO 9 I=1,NCOMP
         X(I,J)=X(I,J)/DENA
       9 CONTINUE
         W(J)=W(J)*DENA
      10 CONTINUE
C              ORGANIC PHASE
         DO 20 J=1,4
         WOM=(4.2-0.015*T(J))*(1.0-2.0*(X(1,J)+X(2,J))/TO-0.6*X(3,J)/TO)*
        1XTBP**1.69
         DENO=1.0+0.097*X(1,J)+0.139*X(2,J)+0.043*X(3,J)+0.0174*WOM
         DO 19 I=1,NCOMP
         X(I,J)=X(I,J)/DENO
      19 CONTINUE
         W(J)=W(J)*DENO
      20 CONTINUE
         RETURN
         END
         SUBROUTINE ORGPH(TIN,XTBP,XOUT,DISCO)
         COMMON/CON/NCOMP,NC5,NE,NS,TMAX,NC3
         DIMENSION XOUT(10),DISCO(10)
C
C           RETURNS A VALUE OF Y/X (DISCO) FOR EACH VALUE OF X TRIED USING
C           SEPHIS DISTRIBUTION DATA
         F=XTBP
         UAM=XOUT(1)
         PUAM=XOUT(2)
         HAM=XOUT(3)
         SNITR=XOUT(4)
         TEMPC=TIN
         IF(UAM.LT.0) UAM=0.0
         IF(PUAM.LT.0) PUAM=0.0
         IF(HAM.LT.0) HAM=0.0
         IF(SNITR.LT.0) SNITR=0.0
         TNM=HAM+2.0*UAM+2.0*PUAM+SNITR
         IF(TNM.EQ.0.0) TNM=1.0
         TEMPRK=1000.0/(TEMPC+273.16)
         UK=3.7*TNM**1.57+1.4*TNM**3.9+0.011*TNM**7.3
         UK=UK*(4.0*F**(-0.17)-3.0)
         PUK=UK*(0.2+0.55*F**1.25+0.0074*TNM**2)
         HK1=0.135*TNM**0.82+0.0052*TNM**3.44
         IF(F.LT.1.0) HK1=HK1*(1.0-0.54*EXP(-15.0*F))
         IF(TEMPC.NE.25.0) UK=UK*EXP(2.5*(TEMPRK-3.3539))
         IF(TEMPC.NE.25.0) PUK=PUK*EXP(-0.2*(TEMPRK-3.3539))
         IF(TEMPC.NE.25.0) HK1=HK1*EXP(0.34*(TEMPRK-3.3539))
         HK2=HK1
         A=2.0*(UK*UAM+PUK*PUAM+HK2*HAM)
         B=HK1*HAM+1.0
         C=-3.65*F
         IF(A.GE.1.0E-6) GO TO 10
         TF=-C/B
         GO TO 15
      10 CONTINUE
         TF=(-B+SQRT(B**2-4.0*A*C))/(2.0*A)
      15 CONTINUE
         DH1=HK1*TF
```

```
      DH2=HK2*TF**2
      DH =DH1+DH2
      DU =UK*TF**2
      DPU=PUK*TF**2
      DISCO(1)=DU
      DISCO(2)=DPU
      DISCO(3)=DH
      DISCO(4)=0.0
      PLTURN
      END
      SUBROUTINE DRIVE (N, T0, H0, Y0, TOUT, EPS, MF, INDEX)
C
C        THIS IS THE DECEMBER 20, 1974 VERSION OF
C        GEAR, A PACKAGE FOR THE SOLUTION OF THE INITIAL VALUE
C        PROBLEM FOR SYSTEMS OF ORDINARY DIFFERENTIAL EQUATIONS,
C                    DY/DT = F(Y,T),    Y = (Y(1),Y(2),...,Y(N)).
C        SUBROUTINE DRIVE IS A DRIVER ROUTINE FOR THE GEAR PACKAGE.
C
C                           REFERENCES
C        1.  A. C. HINDMARSH, GEAR.. ORDINARY DIFFERENTIAL EQUATION
C              SYSTEM SOLVER, UCID-30001 REV. 3, LAWRENCE LIVERMORE
C              LABORATORY, P.O.BOX 808, LIVERMORE, CA 94550, DEC. 1974.
C
C        2.  A. C. HINDMARSH, LINEAR MULTISTEP METHODS FOR ORDINARY
C              DIFFERENTIAL EQUATIONS.. METHOD FORMULATIONS,
C              STABILITY, AND THE METHODS OF NORDSIECK AND GEAR.
C              UCRL-51186 REV. 1, L.L.L., MARCH 1972.
C
C        3.  A. C. HINDMARSH, CONSTRUCTION OF MATHEMATICAL SOFTWARE.
C              PART III.. THE CONTROL OF ERROR IN THE GEAR PACKAGE
C              FOR ORDINARY DIFFERENTIAL EQUATIONS, UCID-30050 PART 3,
C              L.L.L., AUGUST 1972.
C
C-----------------------------------------------------------------------
C  DRIVE IS TO BE CALLED ONCE FOR EACH OUTPUT VALUE OF T, AND
C  IN TURN MAKES REPEATED CALLS TO THE CORE INTEGRATOR, STIFF.
C
C  THE INPUT PARAMETERS ARE..
C    N      =  THE NUMBER OF FIRST-ORDER DIFFERENTIAL EQUATIONS,
C                N CAN BE REDUCED, BUT NEVER INCREASED, DURING A PROBLEM.
C    T0     =  THE INITIAL VALUE OF T. THE INDEPENDENT VARIABLE
C                (USED ONLY ON FIRST CALL).
C    H0     =  THE NEXT STEP SIZE IN T (USED FOR INPUT ONLY ON THE
C                FIRST CALL).
C    Y0     =  A VECTOR OF LENGTH N CONTAINING THE INITIAL VALUES OF
C                Y (USED FOR INPUT ONLY ON FIRST CALL).
C    TOUT   =  THE VALUE OF T AT WHICH OUTPUT IS DESIRED NEXT,
C                INTEGRATION WILL NORMALLY GO SLIGHTLY BEYOND TOUT
C                AND THE PACKAGE WILL INTERPOLATE TO T = TOUT.
C    EPS    =  THE RELATIVE ERROR BOUND  (USED ONLY ON THE
C                FIRST CALL, UNLESS INDEX = -1).  SINGLE STEP ERROR
C                ESTIMATES DIVIDED BY YMAX(I) WILL BE KEPT LESS THAN
C                EPS IN ROOT-MEAN-SQUARE NORM (I.E. EUCLIDEAN NORM
C                DIVIDED BY SQRT(N) ).  THE VECTOR YMAX OF
C                WEIGHTS IS COMPUTED IN DRIVE.  INITIALLY YMAX(I) IS
C                ABS(Y(I)). WITH A DEFAULT VALUE OF 1 IF Y(I) = 0
C                INITIALLY.  THEREAFTER, YMAX(I) IS THE LARGEST VALUE
C                OF ABS(Y(I)) SEEN SO FAR, OR THE INITIAL YMAX(I) IF
C                THAT IS LARGER.  TO ALTER EITHER OF THESE, CHANGE THE
C                APPROPRIATE STATEMENTS IN THE DO-LOOPS ENDING AT
C                STATEMENTS 10 AND 70 BELOW.
C    MF     =  THE METHOD FLAG  (USED ONLY ON FIRST CALL. UNLESS
C                INDEX = -1).  ALLOWED VALUES ARE 10, 11, 12, 13,
C                20, 21, 22, 23.  MF HAS TWO DECIMAL DIGITS, METH
C                AND MITER  (MF = 10*METH + MITER).
C                METH IS THE BASIC METHOD INDICATOR..
C                  METH = 1  MEANS THE ADAMS METHODS,
C                  METH = 2  MEANS THE BACKWARD DIFFERENTIATION
C                            FORMULAS (BDF), OR STIFF METHODS OF GEAR.
C                MITER IS THE ITERATION METHOD INDICATOR..
```

```
C                  MITER = 0 MEANS FUNCTIONAL ITERATION (NO PARTIAL
C                            DERIVATIVES NEEDED).
C                  MITER = 1 MEANS CHORD METHOD WITH ANALYTIC JACOBIAN.
C                            FOR THIS USER SUPPLIES SUBROUTINE
C                            PEDERV  (SEE DESCRIPTION BELOW).
C                  MITER = 2 MEANS CHORD METHOD WITH JACOBIAN CALCULATED
C                            INTERNALLY BY FINITE DIFFERENCES.
C                  MITER = 3 MEANS CHORD METHOD WITH JACOBIAN REPLACED
C                            BY A DIAGONAL APPROXIMATION BASED ON A
C                            DIRECTIONAL DERIVATIVE.
C    INDEX =  INTEGER USED ON INPUT TO INDICATE TYPE OF CALL.
C               WITH THE FOLLOWING VALUES AND MEANINGS..
C                    1     THIS IS THE FIRST CALL FOR THIS PROBLEM.
C                    0     THIS IS NOT THE FIRST CALL FOR THIS PROBLEM,
C                          AND INTEGRATION IS TO CONTINUE.
C                   -1     THIS IS NOT THE FIRST CALL FOR THE PROBLEM,
C                          AND THE USER HAS RESET N, EPS, AND/OR MF.
C                    2     SAME AS 0 EXCEPT THAT TOUT IS TO BE HIT
C                          EXACTLY (NO INTERPOLATION IS DONE).
C                          ASSUMES TOUT .GE. THE CURRENT T.
C                    3     SAME AS 0 EXCEPT CONTROL RETURNS TO CALLING
C                          PROGRAM AFTER ONE STEP.  TOUT IS IGNORED.
C               SINCE THE NORMAL OUTPUT VALUE OF INDEX IS 0,
C               IT NEED NOT BE RESET FOR NORMAL CONTINUATION.
C
C  AFTER THE INITIAL CALL, IF A NORMAL RETURN OCCURRED AND A NORMAL
C  CONTINUATION IS DESIRED, SIMPLY RESET TOUT AND CALL AGAIN.
C  ALL OTHER PARMETERS WILL BE READY FOR THE NEXT CALL.
C  A CHANGE OF PARAMETERS WITH INDEX = -1 CAN BE MADE AFTER
C  EITHER A SUCCESSFUL OR AN UNSUCCESSFUL RETURN.
C
C  THE OUTPUT PARAMETERS ARE..
C    H0     =  THE STEP SIZE H USED LAST, WHETHER SUCCESSFULLY OR NOT.
C    Y0     =  THE COMPUTED VALUES OF Y AT T = TOUT.
C    TOUT   =  THE OUTPUT VALUE OF T.  IF INTEGRATION WAS SUCCESSFUL,
C                AND THE INPUT VALUE OF INDEX WAS NOT 3, TOUT IS
C                UNCHANGED FROM ITS INPUT VALUE.  OTHERWISE, TOUT
C                IS THE CURRENT VALUE OF T TO WHICH INTEGRATION
C                HAS BEEN COMPLETED.
C    INDEX =  INTEGER USED ON OUTPUT TO INDICATE RESULTS,
C                WITH THE FOLLOWING VALUES AND MEANINGS..
C          0     INTEGRATION WAS COMPLETED TO TOUT OR BEYOND.
C         -1     THE INTEGRATION WAS HALTED AFTER FAILING TO PASS THE
C                ERROR TEST EVEN AFTER REDUCING H BY A FACTOR OF
C                1.E10 FROM ITS INITIAL VALUE.
C         -2     AFTER SOME INITIAL SUCCESS, THE INTEGRATION WAS
C                HALTED EITHER BY REPEATED ERROR TEST FAILURES OR BY
C                A TEST ON EPS.  TOO MUCH ACCURACY HAS BEEN REQUESTED.
C         -3     THE INTEGRATION WAS HALTED AFTER FAILING TO ACHIEVE
C                CORRECTOR CONVERGENCE EVEN AFTER REDUCING H BY A
C                FACTOR OF 1.E10 FROM ITS INITIAL VALUE.
C         -4     IMMEDIATE HALT BECAUSE OF ILLEGAL VALUES OF INPUT
C                PARAMETERS.  SEE PRINTED MESSAGE.
C         -5     INDEX WAS -1 ON INPUT, BUT THE DESIRED CHANGES OF
C                PARAMETERS WERE NOT IMPLEMENTED BECAUSE TOUT
C                WAS NOT BEYOND T.  INTERPOLATION TO T = TOUT WAS
C                PERFORMED AS ON A NORMAL RETURN.  TO TRY AGAIN,
C                SIMPLY CALL AGAIN WITH INDEX = -1 AND A NEW TOUT.
C
C   IN ADDITION TO DRIVE, THE FOLLOWING ROUTINES ARE PROVIDED IN
C   THE PACKAGE..
C     INTERP(TOUT,Y,N0,Y0)  INTERPOLATES TO GET THE OUTPUT VALUES
C                           AT T = TOUT, FROM THE DATA IN THE Y ARRAY.
C     STIFF(Y,H0)  IS THE CORE INTEGRATOR ROUTINE.  IT PERFORMS A
C                           SINGLE STEP AND ASSOCIATED ERROR CONTROL.
C     COSET(METH,NQ,EL,TQ,MAXDER)  SETS COEFFICIENTS FOR USE IN
C                           THE CORE INTEGRATOR.
C     PSET(Y,N0,CON,MITER,IER)  COMPUTES AND PROCESSES THE JACOBIAN
C                           MATRIX J = DF/DY.
C     DEC(N,N0,A,IP,IER)  PERFORMS AN LU DECOMPOSITION ON A MATRIX.
C     SOL(N,N0,A,B,IP)  SOLVES LINEAR SYSTEMS AXX = B AFTER DEC
C                           HAS BEEN CALLED FOR THE MATRIX A.
C   NOTE.. PSET, DEC, AND SOL ARE CALLED ONLY IF MITER = 1 OR 2.
```

```
C
C   THE FOLLOWING ROUTINES ARE TO BE SUPPLIED BY THE USER..
C     DIFFUN(N,T,Y,YDOT)  COMPUTES THE FUNCTION YDOT = F(Y,T), THE
C                         RIGHT-HAND SIDE OF THE O.D.E.
C                         HERE Y AND YDOT ARE VECTORS OF LENGTH N.
C     PEDEPY(N,T,Y,PD,N0) COMPUTES THE N BY N JACOBIAN MATRIX OF
C                         PARTIAL DERIVATIVES, AND STORES IT IN PD
C                         AS AN N0 BY N0 ARRAY.  PD(I,J) IS TO BE
C                         SET TO THE PARTIAL DERIVATIVE OF YDOT(I)
C                         WITH RESPECT TO Y(J).  PEDERV IS CALLED
C                         ONLY IF MITER = 1.  OTHERWISE A DUMMY
C                         ROUTINE CAN BE SUBSTITUTED.
C
C  THE DIMENSIONS IN THE FOLLOWING DECLARATIONS ARE SET FOR A
C  MAXIMUM OF 20 EQUATIONS.  IF THE PACKAGE IS TO USED FOR A LARGER
C  VALUE OF N, THE DIMENSIONS SHOULD BE INCREASED ACCORDINGLY.  THE
C  DIMENSION OF PW BELOW MUST BE AT LEAST N**2 IF MITER = 1 OR 2,
C  BUT CAN BE REDUCED TO N IF MITER = 3, OR TO 1 IF MITER = 0.
C  THE DIMENSIONS OF YMAX, ERROR, SAVE1, SAVE2, IPIV, AND THE FIRST
C  DIMENSION OF Y SHOULD ALL BE AT LEAST N.  THE COLUMN LENGTH OF
C  THE Y ARRAY AS USED ELSEWHERE IS N0, NOT 20.  THE ROW LENGTH OF Y
C  CAN BE REDUCED FROM 13 TO 6 IF METH = 2.
C  THE IPIV ARRAY IS USED ONLY IF MITER IS 1 OR 2.
C
C  THE COMMON BLOCK GEAR9 CAN BE ACCESSED EXTERNALLY BY THE USER
C  IF DESIRED.  IT CONTAINS THE STEP SIZE LAST USED (SUCCESSFULLY),
C  THE ORDER LAST USED (SUCCESSFULLY), THE NUMBER OF STEPS TAKEN
C  SO FAR, THE NUMBER OF F EVALUATIONS (DIFFUN CALLS) SO FAR,
C  AND THE NUMBER OF JACOBIAN EVALUATIONS SO FAR.
C
C  IN THE FOLLOWING DATA STATEMENT, SET..
C    UROUND    THE UNIT ROUNDOFF OF THE MACHINE, I.E. THE SMALLEST
C              POSITIVE U SUCH THAT 1. + U .NE. 1. ON THE MACHINE.
C    LOUT   =  THE LOGICAL UNIT NUMBER FOR THE OUTPUT OF MESSAGES
C              DURING THE INTEGRATION.
C-----------------------------------------------------------------------
C
C+----------------------------------------------------------------------+CC
C+                                                                      +CC
C+                        LAWRENCE LIVERMORE LABORATORY                 +CC
C+      NUMERICAL MATHEMATICS GROUP -- MATHEMATICAL SOFTWARE LIBRARY    +CC
C+                                                                      +CC
C+----------------------------------------------------------------------+CC
C+                                                                      +CC
C+                  CLASS ONE ROUTINE:  DRIVE                           +CC
C+                           REVISION:  0                               +CC
C+                  DATE LAST CHANGED:  78-02-10                        +CC
C+                     RELEASE STATUS:  UNLIMITED                       +CC
C+                                                                      +CC
C+  EACH  CLASS ONE  ROUTINE HAS BEEN THOROUGHLY TESTED BY NMG AND MEETS +CC
C+  CERTAIN DOCUMENTATION AND PROGRAMMING STANDARDS.                     +CC
C+                                                                      +CC
C+  AT LEAST ONE CONSULTANT IS AVAILABLE TO ANSWER QUESTIONS AND RESPOND +CC
C+  TO REPORTED ERRORS OR INADEQUACIES IN A  CLASS ONE  ROUTINE.        +CC
C+                                                                      +CC
C+  +------------------------------------------------------------------+CC
C+  +                          N O T I C E                             +CC
C+  +                                                                  +CC
C+  +  THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE +CC
C+  +  UNITED STATES GOVERNMENT.  NEITHER THE UNITED STATES NOR THE    +CC
C+  +  UNITED STATES ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION,   +CC
C+  +  NOR ANY OF THEIR EMPLOYEES, NOR ANY OF THEIR CONTRACTORS, SUB-  +CC
C+  +  CONTRACTORS, OR THEIR EMPLOYEES, MAKES ANY WARRENTY, EXPRESS OR +CC
C+  +  IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR   +CC
C+  +  THE ACCURACY, COMPLETENESS OR USEFULNESS OF ANY INFORMATION,    +CC
C+  +  APPARATUS, PRODUCT OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS +CC
C+  +  USE WOULD NOT INFRINGE PRIVATELY-OWNED RIGHTS.                  +CC
C+  +                                                                  +CC
C+  +------------------------------------------------------------------+CC
C+                                                                      +CC
C+  PLEASE REPORT ANY SUSPECTED ERRORS IN THIS ROUTINE IMMEDIATELY TO NMG.+CC
C+  EXT. 3049, 3329, OR 3288.                                           +CC
C+                                                                      +CC
C+----------------------------------------------------------------------+CC
```

```
2186 C
2187       DIMENSION YDOT(300)
2188       DIMENSION Y0(N)
2189       DIMENSION Y(300,13)
2190       LCM /GEAP6/
2191       COMMON /GEAR1/ T,H,HMIN,HMAX,EPSC,UROUND,NC,MFC,KFLAG,JSTART
2192       COMMON /GEAP2/ YMAX(300)
2193       COMMON /GEAP3/ EPPOP(300)
2194       COMMON /GEAP4/ SAVE1(300)
2195       COMMON /GEAP5/ SAVE2(300)
2196       COMMON /GEAP6/ PW(90000)
2197       COMMON /GEAP7/ IPIV(300)
2198       COMMON /GEAP8/ EPSJ,NSQ
2199       COMMON /GEAP9/ NUSED,NOUSED,NSTEP,NFE,NJE
2200       COMMON /LARM/ STD,PPCNT,EMC,EMSTD,EMPRCT
2201       DATA UROUND/7.1E-15/, LOUT/3/
2202 C
2203       IF (INDEX .EQ. 0) GO TO 20
2204       IF (INDEX .EQ. 2) GO TO 25
2205       IF (INDEX .EQ. -1) GO TO 30
2206       IF (INDEX .EQ. 3) GO TO 40
2207       IF (INDEX .NE. 1) GO TO 430
2208       IF (EPS .LE. 0.) GO TO 480
2209       IF (N .LE. 0) GO TO 410
2210       IF ((T0-TOUT)*H0 .GE. 0.) GO TO 420
2211 C---------- -  --- - ------------------------------------------------------
2212 C  IF INITIAL VALUES OF YMAX OTHER THAN THOSE SET BELOW ARE DESIRED,
2213 C  THEY SHOULD BE SET HERE.  ALL YMAX(I) MUST BE POSITIVE.
2214 C  IF VALUES FOR HMIN OR HMAX, THE BOUNDS ON ABS(H), OTHER THAN
2215 C  THOSE BELOW ARE DESIRED, THEY SHOULD BE SET BELOW.
2216 C----- ---- ---- ---- -  ------------------------------------------------
2217       DO 10 I = 1,N
2218         YMAX(I) = ABS(Y0(I))
2219         IF (YMAX(I) .EQ. 0.) YMAX(I) = 1.
2220  10     Y(1,I) = Y0(I)
2221       NC = 0
2222       T = T0
2223       H = H0
2224       IF ((T+H) .EQ. T) WRITE(LOUT,15)
2225  15   FORMAT(35H WARNING..  T + H = T ON NEXT STEP.)
2226       HMIN = ABS(H0)
2227       HMAX = ABS(T0-TOUT)*10.
2228       EPSC = EPS
2229       MFC = MF
2230       JSTART = 0
2231       N0 = N
2232       NSQ = N0*N0
2233       EPSJ = SQRT(UROUND)
2234       NHCUT = 0
2235       GO TO 50
2236 C
2237 C  TOUTP IS THE PREVIOUS VALUE OF TOUT FOR USE IN HMAX.-----------------
2238  20   HMAX = ABS(TOUT-TOUTP)*10.
2239       GO TO 80
2240 C
2241  25   HMAX = ABS(TOUT-TOUTP)*10.
2242       IF ((T-TOUT)*H .GE. 0.) GO TO 500
2243       GO TO 85
2244 C
2245  30   IF ((T-TOUT)*H .GE. 0.) GO TO 440
2246       JSTART = -1
2247       NC = N
2248       EPSC = EPS
2249       MFC = MF
2250 C
2251  40   IF ((T+H) .EQ. T) WRITE(LOUT,15)
2252 C
2253  50   CALL STIFF (Y, N0)
2254 C
2255       KGO = 1 - KFLAG
2256       GO TO (60, 100, 200, 300), KGO
2257 C  KFLAG =   0.  -1.  -2.  -3
2258 C
```

```
      60   CONTINUE
C-----------------------------------------------------------------------
C  NORMAL RETURN FROM INTEGRATOR.
C
C  THE WEIGHTS YMAX(I) ARE UPDATED.  IF DIFFERENT VALUES ARE DESIRED,
C  THEY SHOULD BE SET HERE.  A TEST IS MADE FOR EPS BEING TOO SMALL
C  FOR THE MACHINE PRECISION.
C
C  ANY OTHER TESTS OR CALCULATIONS THAT ARE REQUIRED AFTER EVERY
C  STEP SHOULD BE INSERTED HERE.
C
C  IF INDEX = 3, Y0 IS SET TO THE CURRENT Y VALUES ON RETURN.
C  IF INDEX = 2, H IS CONTROLLED TO HIT TOUT (WITHIN ROUNDOFF
C  ERROR), AND THEN THE CURRENT Y VALUES ARE PUT IN Y0 ON RETURN.
C  FOR ANY OTHER VALUE OF INDEX, CONTROL RETURNS TO THE INTEGRATOR
C  UNLESS TOUT HAS BEEN REACHED.  THEN INTERPOLATED VALUES OF Y ARE
C  COMPUTED AND STORED IN Y0 ON RETURN.
C  IF INTERPOLATION IS NOT DESIRED, THE CALL TO INTERP SHOULD BE
C  REMOVED AND CONTROL TRANSFERRED TO STATEMENT 500 INSTEAD OF 520.
C-----------------------------------------------------------------------
      D = 0.
      DO 70 I = 1,N
        AYI = ABS(Y(I,1))
        YMAX(I) = AMAX1(YMAX(I), AYI)
   70   D = D + (AYI/YMAX(I))**2
      D = D*(UROUND/EPS)**2
      IF (D .GT. FLOAT(N)) GO TO 250
      IF (INDEX .EQ. 3) GO TO 500
      IF (INDEX .EQ. 2) GO TO 85
   80 IF ((T-TOUT)*H .LT. 0.) GO TO 40
      CALL INTERP (TOUT, Y, N0, Y0)
      GO TO 520
   85 IF (((T+H)-TOUT)*H .LE. 0.) GO TO 40
      IF (ABS(T-TOUT) .LE. 100.*UROUND*HMAX) GO TO 500
      IF ((T-TOUT)*H .GE. 0.) GO TO 500
      H = (TOUT - T)*(1. - 4.*UROUND)
      JSTART = -1
      GO TO 40
C-----------------------------------------------------------------------
C  ON AN ERROR RETURN FROM INTEGRATOR, AN IMMEDIATE RETURN OCCURS IF
C  KFLAG = -2, AND RECOVERY ATTEMPTS ARE MADE OTHERWISE.
C  TO RECOVER, H AND HMIN ARE REDUCED BY A FACTOR OF .1 UP TO 10
C  TIMES BEFORE GIVING UP.
C-----------------------------------------------------------------------
  100 WRITE (LOUT,105) T
  105 FORMAT(//35H KFLAG = -1 FROM INTEGRATOR AT T = ,E16.8/
     1   38H  ERROR TEST FAILED WITH ABS(H) = HMIN/)
  110 IF (NHCUT .EQ. 10) GO TO 150
      NHCUT = NHCUT + 1
      HMIN = .1*HMIN
      H = .1*H
      WRITE (LOUT,115) H
  115 FORMAT(24H  H HAS BEEN REDUCED TO ,E16.8,
     1   26H  AND STEP WILL BE RETRIED//)
      JSTART = -1
      GO TO 40
C
  150 WRITE (LOUT,155)
  155 FORMAT(//44H PROBLEM APPEARS UNSOLVABLE WITH GIVEN INPUT//)
      GO TO 500
C
  200 WRITE (LOUT,205) T,H
  205 FORMAT(//35H KFLAG = -2 FROM INTEGRATOR AT T = ,E16.8,5H  H =,
     1  E16.8/52H  THE REQUESTED ERROR IS SMALLER THAN CAN BE HANDLED//)
      GO TO 500
C
  250 WRITE (LOUT,255) T
  255 FORMAT(//37H INTEGRATION HALTED BY DRIVER AT T = ,E16.8/
     1   56H  EPS TOO SMALL TO BE ATTAINED FOR THE MACHINE PRECISION/)
      KFLAG = -2
      GO TO 500
C
  300 WRITE (LOUT,305) T
```

```
2332   305  FORMAT(//35H KFLAG = -3 FROM INTEGRATOR AT T = ,E16.8/
2333      1   45H  CORRECTOR CONVERGENCE COULD NOT BE ACHIEVED/)
2334        GO TO 110
2335  C
2336   400  WRITE (LOUT,405)
2337   405  FORMAT(//28H ILLEGAL INPUT.. EPS .LE. 0.//)
2338        INDEX = -4
2339        RETURN
2340  C
2341   410  WRITE (LOUT,415)
2342   415  FORMAT(//25H ILLEGAL INPUT.. N .LE. 0//)
2343        INDEX = -4
2344        RETURN
2345  C
2346   420  WRITE (LOUT,425)
2347   425  FORMAT(//36H ILLEGAL INPUT.. (T0-TOUT)*H .GE. 0.//)
2348        INDEX = -4
2349        RETURN
2350  C
2351   430  WRITE (LOUT,435) INDEX
2352   435  FORMAT(//24H ILLEGAL INPUT.. INDEX =,I5//)
2353        INDEX = -4
2354        RETURN
2355  C
2356   440  WRITE(LOUT,445) T,TOUT,H
2357   445  FORMAT(//44H INDEX = -1 ON INPUT WITH (T-TOUT)*H .GE. 0./
2358      1   4H T =,E16.8,9H   TOUT =,E16.8,6H   H =,E16.8/
2359      1   44H INTERPOLATION WAS DONE AS ON NORMAL RETURN./
2360      2   41H DESIRED PARAMETER CHANGES WERE NOT MADE.)
2361        CALL INTERP (TOUT, Y, N0, Y0)
2362        INDEX = -5
2363        RETURN
2364  C
2365   500  TOUT = T
2366        DO 510 I = 1,N
2367   510    Y0(I) = Y(I,1)
2368   520  INDEX = KFLAG
2369        TOUTP = TOUT
2370        H0 = HUSED
2371        IF (KFLAG .NE. 0) H0 = H
2372        CALL NOISE (N,Y0,STD,PRCNT,EMC)
2373        RETURN
2374  C----------------------- END OF SUBROUTINE DRIVE ------------------------
2375        END
2376  C
2377        SUBROUTINE NOISE (N,Y0,STD,PRCNT,EMC)
2378  C
2379  C       THIS SUBROUTINE COMPUTES AND ADDS NOISE TO THE VARIABLE VECTOR
2380  C       WHENEVER CALLED
2381  C
2382  C           STD IS THE ABSOLUTE STANDARD DEVIATION OF THE NOISE
2383  C           PRCNT IS THE PER CENT OF VALUE STANDARD DEVIATION OF THE NOISE
2384  C           STD OR PRCNT SHOULD BE ZERO
2385  C
2386  C           RNFL IS THE RANDOM NUMBER GENERATOR - IT IS MACHINE DEPENDENT
2387  C
2388        DIMENSION Y0(N)
2389        IF (STD.EQ.0.0.AND.PRCNT.EQ.0.0) RETURN
2390        IF (STD.EQ.0.0) GO TO 10
2391        IF (STD.NE.0.0.AND.PRCNT.NE.0.0) WRITE(3,100)
2392        IF (STD.NE.0.0.AND.PRCNT.NE.0.0) PRCNT=0.0
2393        DO 2 J=1,N
2394        GAUSS=0.0
2395        DO 1 I=1,12
2396        GAUSS=GAUSS+RNFL(EMC)
2397        CALL IRNFL(EMC)
2398      1 CONTINUE
2399        GAUSS=(GAUSS-6.0)*STD
2400        Y0(J)=Y0(J)+GAUSS
2401      2 CONTINUE
2402        GO TO 20
2403     10 DO 12 J=1,N
2404        GAUSS=0.0
```

```
      DO 11 I=1,12
      GAUSS=GAUSS+RNFL(EMC)
      CALL RNFL(EMC)
   11 CONTINUE
      GAUSS=(GAUSS-6.0)*PRCNT*Y0(J)/100.0
      Y0(J)=Y0(J)+GAUSS
   12 CONTINUE
   20 CONTINUE
      RETURN
  100 FORMAT (10X,"*** EITHER STD OR PRCNT SHOULD BE ZERO, STD WAS USED ***")
      END

C
C
      SUBROUTINE STIFF (Y, N0)
C-----------------------------------------------------------------------
C   THE FOLLOWING CARD IS FOR OPTIMIZED COMPILATION UNDER CHAT.
      OPTIMIZE
C-----------------------------------------------------------------------
      DIMENSION Y(N0,1)
      LCM (GEAR6)
      COMMON /GEAR1/ T,H,HMIN,HMAX,EPS,UROUND,N,MF,KFLAG,JSTART
      COMMON /GEAR2/ YMAX(1)
      COMMON /GEAR3/ ERROR(1)
      COMMON /GEAR4/ SAVE1(1)
      COMMON /GEAR5/ SAVE2(1)
      COMMON /GEAR6/ PW(1)
      COMMON /GEAR7/ IPIV(1)
      COMMON /GEAR9/ HUSED,NQUSED,HSTEP,NFE,NJE
C-----------------------------------------------------------------------
C  STIFF PERFORMS ONE STEP OF THE INTEGRATION OF AN INITIAL VALUE
C  PROBLEM FOR A SYSTEM OF ORDINARY DIFFERENTIAL EQUATIONS.
C  COMMUNICATION WITH STIFF IS DONE WITH THE FOLLOWING VARIABLES..
C
C     Y       AN N0 BY LMAX ARRAY CONTAINING THE DEPENDENT VARIABLES
C               AND THEIR SCALED DERIVATIVES.  LMAX IS 13 FOR THE ADAMS
C               METHODS AND 6 FOR THE BDF METHODS.  LMAX - 1 = MAXDER
C               IS THE MAXIMUM ORDER AVAILABLE.  SEE SUBROUTINE COSET.
C               Y(I,J+1) CONTAINS THE J-TH DERIVATIVE OF Y(I), SCALED BY
C               H**J/FACTORIAL(J)  (J = 0,1,...,NQ).
C     N0      A CONSTANT INTEGER .GE. N, USED FOR DIMENSIONING PURPOSES.
C     T       THE INDEPENDENT VARIABLE.  T IS UPDATED ON EACH STEP TAKEN.
C     H       THE STEP SIZE TO BE ATTEMPTED ON THE NEXT STEP.
C               H IS ALTERED BY THE ERROR CONTROL ALGORITHM DURING THE
C               PROBLEM.  H CAN BE EITHER POSITIVE OR NEGATIVE, BUT ITS
C               SIGN MUST REMAIN CONSTANT THROUGHOUT THE PROBLEM.
C     HMIN,   THE MINIMUM AND MAXIMUM ABSOLUTE VALUE OF THE STEP SIZE
C      HMAX     TO BE USED FOR THE STEP.  THESE MAY BE CHANGED AT ANY
C               TIME, BUT WILL NOT TAKE EFFECT UNTIL THE NEXT H CHANGE.
C     EPS     THE RELATIVE ERROR BOUND.  SEE DESCRIPTION IN DRIVER.
C     UROUND  THE UNIT ROUNDOFF OF THE MACHINE.
C     N       THE NUMBER OF FIRST-ORDER DIFFERENTIAL EQUATIONS.
C     MF      THE METHOD FLAG.  SEE DESCRIPTION IN DRIVER.
C     KFLAG   A COMPLETION CODE WITH THE FOLLOWING MEANINGS..
C                         0  THE STEP WAS SUCCESFUL.
C                        -1  THE REQUESTED ERROR COULD NOT BE ACHIEVED
C                            WITH ABS(H) = HMIN.
C                        -2  THE REQUESTED ERROR IS SMALLER THAN CAN
C                            BE HANDLED FOR THIS PROBLEM.
C                        -3  CORRECTOR CONVERGENCE COULD NOT BE
C                            ACHIEVED FOR ABS(H) = HMIN.
C               ON A RETURN WITH KFLAG NEGATIVE, THE VALUES OF T AND
C               THE Y ARRAY ARE AS OF THE BEGINNING OF THE LAST
C               STEP, AND H IS THE LAST STEP SIZE ATTEMPTED.
C     JSTART  AN INTEGER USED ON INPUT AND OUTPUT.
C               ON INPUT, IT HAS THE FOLLOWING VALUES AND MEANINGS..
C                    0  PERFORM THE FIRST STEP.
C                 .GT.0  TAKE A NEW STEP CONTINUING FROM THE LAST.
C                 .LT.0  TAKE THE NEXT STEP WITH A NEW VALUE OF
C                            H, EPS, N, AND/OR MF.
C               ON EXIT, JSTART IS NQ, THE CURRENT ORDER OF THE METHOD.
C     YMAX    AN ARRAY OF N ELEMENTS WITH WHICH THE ESTIMATED LOCAL
```

```
C                ERRORS IN Y ARE COMPARED.
C     ERROR    AN ARRAY OF N ELEMENTS.  ERROR(I)/TQ(2) IS THE ESTIMATED
C                ONE-STEP ERROR IN Y(I).
C     SAVE1,   TWO ARRAYS OF WORKING STORAGE,
C      SAVE2     EACH OF LENGTH N.
C     PW       A BLOCK OF LOCATIONS USED FOR PARTIAL DERIVATIVES IF
C                MITER IS NOT 0.  SEE DESCRIPTION IN DRIVER.
C     IPIV     AN INTEGER ARRAY OF LENGTH N USED FOR PIVOT
C                INFORMATION IF MITER = 1 OR 2.
C-----------------------------------------------------------------------
      DIMENSION EL(13),TQ(4)
      DATA EL(2)/1./, OLDL0/1./
      KFLAG = 0
      TOLD = T
      IF (JSTART .GT. 0) GO TO 200
      IF (JSTART .NE. 0) GO TO 120
C-----------------------------------------------------------------------
C  ON THE FIRST CALL, THE ORDER IS SET TO 1 AND THE INITIAL YDOT IS
C  CALCULATED.  RMAX IS THE MAXIMUM RATIO BY WHICH H CAN BE INCREASED
C  IN A SINGLE STEP.  IT IS INITIALLY 1.E4 TO COMPENSATE FOR THE SMALL
C  INITIAL H, BUT THEN IS NORMALLY EQUAL TO 10.  IF A FAILURE
C  OCCURS (IN CORRECTOR CONVERGENCE OR ERROR TEST), RMAX IS SET AT 2
C  FOR THE NEXT INCREASE.
C-----------------------------------------------------------------------
      CALL DYN2 (N, T, Y, SAVE1)
      DO 110 I = 1,N
  110   Y(I,2) = H*SAVE1(I)
      METH = MF/10
      MITER = MF - 10*METH
      NQ = 1
      L = 2
      IDOUB = 3
      RMAX = 1.E4
      RC = 0.
      CRATE = 1.
      HOLD = H
      MFOLD = MF
      NSTEP = 0
      NSTEPJ = 0
      NFE = 1
      NJE = 0
      IRET = 3
      GO TO 130
C-----------------------------------------------------------------------
C  IF THE CALLER HAS CHANGED METH, CDSET IS CALLED TO SET
C  THE COEFFICIENTS OF THE METHOD.  IF THE CALLER HAS CHANGED
C  N, EPS, OR METH, THE CONSTANTS E, EDN, EUP, AND BND MUST BE RESET.
C  E IS A COMPARISON FOR ERRORS OF THE CURRENT ORDER NQ. EUP IS
C  TO TEST FOR INCREASING THE ORDER, EDN FOR DECREASING THE ORDER.
C  BND IS USED TO TEST FOR CONVERGENCE OF THE CORRECTOR ITERATES.
C  IF THE CALLER HAS CHANGED H, Y MUST BE RESCALED.
C  IF H OR METH HAS BEEN CHANGED, IDOUB IS RESET TO L + 1 TO PREVENT
C  FURTHER CHANGES IN H FOR THAT MANY STEPS.
C-----------------------------------------------------------------------
  120 IF (MF .EQ. MFOLD) GO TO 150
      MEO = METH
      MIO = MITER
      METH = MF/10
      MITER = MF - 10*METH
      MFOLD = MF
      IF (MITER .NE. MIO) IWEVAL = MITER
      IF (METH .EQ. MEO) GO TO 150
      IDOUB = L + 1
      IRET = 1
  130 CALL CDSET (METH, NQ, EL, TQ, MAXDER)
      LMAX = MAXDER + 1
      RC = RC*EL(1)/OLDL0
      OLDL0 = EL(1)
  140 FN = FLOAT(N)
      EDN = FN*(TQ(1)*EPS)**2
      E   = FN*(TQ(2)*EPS)**2
      EUP = FN*(TQ(3)*EPS)**2
      BND = FN*(TQ(4)*EPS)**2
```

```
2549        EPSOLD = EPS
2550        NOLD = N
2551        GO TO (160, 170, 200), IRET
2552   150  IF ((EPS .EQ. EPSOLD) .AND. (N .EQ. NOLD)) GO TO 160
2553        IF (N .NE. NOLD) IWEVAL = MITER
2554        IRET = 1
2555        GO TO 140
2556   160  IF (H .EQ. HOLD) GO TO 200
2557        RH = H/HOLD
2558        H = HOLD
2559        IREDO = 3
2560        GO TO 175
2561   170  RH = AMAX1(RH,HMIN/ABS(H))
2562   175  RH = AMIN1(RH,HMAX/ABS(H),RMAX)
2563        R1 = 1.
2564        DO 180 J = 2,L
2565          R1 = R1*RH
2566          DO 180 I = 1,N
2567   180      Y(I,J) = Y(I,J)*R1
2568        H = H*RH
2569        RC = RC*RH
2570        IDOUB = L + 1
2571        IF (IREDO .EQ. 0) GO TO 690
2572  C-------------------------------------------------------------------
2573  C  THIS SECTION COMPUTES THE PREDICTED VALUES BY EFFECTIVELY
2574  C  MULTIPLYING THE Y ARRAY BY THE PASCAL TRIANGLE MATRIX.
2575  C  RC IS THE RATIO OF NEW TO OLD VALUES OF THE COEFFICIENT  H*EL(1).
2576  C  WHEN RC DIFFERS FROM 1 BY MORE THAN 30 PERCENT, OR THE CALLER HAS
2577  C  CHANGED MITER, IWEVAL IS SET TO MITER TO FORCE THE PARTIALS TO BE
2578  C  UPDATED, IF PARTIALS ARE USED.  IN ANY CASE, THE PARTIALS
2579  C  ARE UPDATED AT LEAST EVERY 20-TH STEP.
2580  C-------------------------------------------------------------------
2581   200  IF (ABS(RC-1.) .GT. 0.3) IWEVAL = MITER
2582        IF (NSTEP .GE. NSTEPJ+20) IWEVAL = MITER
2583        T = T + H
2584        DO 210 J1 = 1,NQ
2585          DO 210 J2 = J1,NQ
2586            J = (NQ + J1) - J2
2587            DO 210 I = 1,N
2588   210        Y(I,J) = Y(I,J) + Y(I,J+1)
2589  C-------------------------------------------------------------------
2590  C  UP TO 3 CORRECTOR ITERATIONS ARE TAKEN.  A CONVERGENCE TEST IS
2591  C  MADE ON THE R.M.S. NORM OF EACH CORRECTION, USING BND, WHICH
2592  C  IS DEPENDENT ON EPS.  THE SUM OF THE CORRECTIONS IS ACCUMULATED
2593  C  IN THE VECTOR ERROR(I).  THE Y ARRAY IS NOT ALTERED IN THE CORRECTOR
2594  C  LOOP.  THE UPDATED Y VECTOR IS STORED TEMPORARILY IN SAVE1.
2595  C-------------------------------------------------------------------
2596   220  DO 230 I = 1,N
2597   230    ERROR(I) = 0.
2598        M = 0
2599        CALL DYN2 (N, T, Y, SAVE2)
2600        NFE = NFE + 1
2601        IF (IWEVAL .LE. 0) GO TO 290
2602  C-------------------------------------------------------------------
2603  C  IF INDICATED, THE MATRIX P = I - H*EL(1)*J IS REEVALUATED BEFORE
2604  C  STARTING THE CORRECTOR ITERATION.  IWEVAL IS SET TO 0 AS AN
2605  C  INDICATOR THAT THIS HAS BEEN DONE.  IF MITER = 1 OR 2, P IS
2606  C  COMPUTED AND PROCESSED IN PSET.  IF MITER = 3, THE MATRIX USED
2607  C  IS P = I - H*EL(1)*D, WHERE D IS A DIAGONAL MATRIX.
2608  C-------------------------------------------------------------------
2609        IWEVAL = 0
2610        RC = 1.
2611        NJE = NJE + 1
2612        NSTEPJ = NSTEP
2613        GO TO (250, 240, 260), MITER
2614   240  NFE = NFE + N
2615   250  CON = -H*EL(1)
2616        CALL PSET (Y, NQ, CON, MITER, IER)
2617        IF (IER .NE. 0) GO TO 420
2618        GO TO 350
2619   260  R = EL(1)*.1
2620        DO 270 I = 1,N
2621   270    PW(I) = Y(I,1) + R*(H*SAVE2(I) - Y(I,2))
2622        CALL DYN2 (N, T, PW, SAVE1)
```

```
      NFE = NFE + 1
      HL0 = H*EL(1)
      DO 280 I = 1,N
        R0 = H*SAVE2(I) - Y(I,2)
        PW(I) = 1.
        D = .1*R0 - H*(SAVE1(I) - SAVE2(I))
        SAVE1(I) = 0.
        IF (ABS(R0) .LT. UROUND*YMAX(I)) GO TO 280
        IF (ABS(D) .EQ. 0.) GO TO 420
        PW(I) = .1*R0/D
        SAVE1(I) = PW(I)*R0
 280    CONTINUE
      GO TO 370
 290  IF (MITER .NE. 0) GO TO (350, 350, 310), MITER
C-----------------------------------------------------------------------
C  IN THE CASE OF FUNCTIONAL ITERATION, UPDATE Y DIRECTLY FROM
C  THE RESULT OF THE LAST DIFFUN CALL.
C-----------------------------------------------------------------------
      D = 0.
      DO 300 I = 1,N
        R = H*SAVE2(I) - Y(I,2)
        D = D + ( (R-ERROR(I))/YMAX(I) )**2
        SAVE1(I) = Y(I,1) + EL(1)*R
 300    ERROR(I) = R
      GO TO 400
C-----------------------------------------------------------------------
C  IN THE CASE OF THE CHORD METHOD, COMPUTE THE CORRECTOR ERROR,
C  F SUB (M), AND SOLVE THE LINEAR SYSTEM WITH THAT AS RIGHT-HAND
C  SIDE AND P AS COEFFICIENT MATRIX, USING THE LU DECOMPOSITION
C  IF MITER = 1 OR 2.  IF MITER = 3, THE COEFFICIENT H*EL(1)
C  IN P IS UPDATED.
C-----------------------------------------------------------------------
 310  PHL0 = HL0
      HL0 = H*EL(1)
      IF (HL0 .EQ. PHL0) GO TO 330
      R = HL0/PHL0
      DO 320 I = 1,N
        D = 1. - R*(1. - 1./PW(I))
        IF (ABS(D) .EQ. 0.) GO TO 440
 320    PW(I) = 1./D
 330  DO 340 I = 1,N
 340    SAVE1(I) = PW(I)*(H*SAVE2(I) - (Y(I,2) + ERROR(I)))
      GO TO 370
 350  DO 360 I = 1,N
 360    SAVE1(I) = H*SAVE2(I) - (Y(I,2) + ERROR(I))
      CALL SOL (N, NQ, PW, SAVE1, IPIV)
 370  D = 0.
      DO 380 I = 1,N
        ERROR(I) = ERROR(I) + SAVE1(I)
        D = D + (SAVE1(I)/YMAX(I))**2
 380    SAVE1(I) = Y(I,1) + EL(1)*ERROR(I)
C-----------------------------------------------------------------------
C  TEST FOR CONVERGENCE.  IF M.GT.0, AN ESTIMATE OF THE CONVERGENCE
C  RATE CONSTANT IS STORED IN CRATE, AND THIS IS USED IN THE TEST.
C-----------------------------------------------------------------------
 400  IF (M .NE. 0) CRATE = AMAX1(.9*CRATE,D/D1)
      IF ((D*AMIN1(1.,2.*CRATE)) .LE. BND) GO TO 450
      D1 = D
      M = M + 1
      IF (M .EQ. 3) GO TO 410
      CALL DYN2 (N, T, SAVE1, SAVE2)
      GO TO 290
C-----------------------------------------------------------------------
C  THE CORRECTOR ITERATION FAILED TO CONVERGE IN 3 TRIES.  IF PARTIALS
C  ARE INVOLVED BUT ARE NOT UP TO DATE, THEY ARE REEVALUATED FOR THE
C  NEXT TRY.  OTHERWISE THE Y ARRAY IS RETRACTED TO ITS VALUES
C  BEFORE PREDICTION, AND H IS REDUCED, IF POSSIBLE.  IF NOT, A
C  NO-CONVERGENCE EXIT IS TAKEN.
C-----------------------------------------------------------------------
 410  NFE = NFE + 2
      IF (IUEVAL .EQ. -1) GO TO 440
 420  T = TOLD
      RMAX = 2.
```

```
      DO 430 J1 = 1,NQ
        DO 430 J2 = J1,NQ
          J = (NQ + J1) - J2
          DO 430 I = 1,N
430         Y(I,J) = Y(I,J) - Y(I,J+1)
      IF (ABS(H) .LE. HMIN*1.00001) GO TO 680
      RH = .25
      IREDO = 1
      GO TO 170
440   IWEVAL = MITER
      GO TO 220
C-----------------------------------------------------------------------
C  THE CORRECTOR HAS CONVERGED.  IWEVAL IS SET TO -1 IF PARTIAL
C  DERIVATIVES WERE USED, TO SIGNAL THAT THEY MAY NEED UPDATING ON
C  SUBSEQUENT STEPS.  THE ERROR TEST IS MADE AND CONTROL PASSES TO
C  STATEMENT 500 IF IT FAILS.
C-----------------------------------------------------------------------
450   IF (MITER .NE. 0) IWEVAL = -1
      NFE = NFE + M
      D = 0.
      DO 460 I = 1,N
460     D = D + (ERROR(I)/YMAX(I))**2
      IF (D .GT. E) GO TO 500
C-----------------------------------------------------------------------
C  AFTER A SUCCESSFUL STEP, UPDATE THE Y ARRAY.
C  CONSIDER CHANGING H IF IDOUB = 1.  OTHERWISE DECREASE IDOUB BY 1.
C  IF IDOUB IS THEN 1 AND NQ .LT. MAXDER, THEN ERROR IS SAVED FOR
C  USE IN A POSSIBLE ORDER INCREASE ON THE NEXT STEP.
C  IF A CHANGE IN H IS CONSIDERED, AN INCREASE OR DECREASE IN ORDER
C  BY ONE IS CONSIDERED ALSO.  A CHANGE IN H IS MADE ONLY IF IT IS BY A
C  FACTOR OF AT LEAST 1.1.  IF NOT, IDOUB IS SET TO 10 TO PREVENT
C  TESTING FOR THAT MANY STEPS.
C-----------------------------------------------------------------------
      KFLAG = 0
      IREDO = 0
      NSTEP = NSTEP + 1
      HUSED = H
      NQUSED = NQ
      DO 470 J = 1,L
        DO 470 I = 1,N
470       Y(I,J) = Y(I,J) + EL(J)*ERROR(I)
      IF (IDOUB .EQ. 1) GO TO 520
      IDOUB = IDOUB - 1
      IF (IDOUB .GT. 1) GO TO 700
      IF (L .EQ. LMAX) GO TO 700
      DO 490 I = 1,N
490     Y(I,LMAX) = ERROR(I)
      GO TO 700
C-----------------------------------------------------------------------
C  THE ERROR TEST FAILED.  KFLAG KEEPS TRACK OF MULTIPLE FAILURES.
C  RESTORE T AND THE Y ARRAY TO THEIR PREVIOUS VALUES, AND PREPARE
C  TO TRY THE STEP AGAIN.  COMPUTE THE OPTIMUM STEP SIZE FOR THIS OR
C  ONE LOWER ORDER.
C-----------------------------------------------------------------------
500   KFLAG = KFLAG - 1
      T = TOLD
      DO 510 J1 = 1,NQ
        DO 510 J2 = J1,NQ
          J = (NQ + J1) - J2
          DO 510 I = 1,N
510         Y(I,J) = Y(I,J) - Y(I,J+1)
      RMAX = 2.
      IF (ABS(H) .LE. HMIN*1.00001) GO TO 660
      IF (KFLAG .LE. -3) GO TO 640
      IREDO = 2
      RH3 = 1.E+20
      GO TO 540
C-----------------------------------------------------------------------
C  REGARDLESS OF THE SUCCESS OR FAILURE OF THE STEP, FACTORS
C  RH1, RH2, AND RH3 ARE COMPUTED, BY WHICH H COULD BE DIVIDED
C  AT ORDER NQ - 1, ORDER NQ, OR ORDER NQ + 1, RESPECTIVELY.
C  IN THE CASE OF FAILURE, RH3 = 1.E20 TO AVOID AN ORDER INCREASE.
C  THE SMALLEST OF THESE IS DETERMINED AND THE NEW ORDER CHOSEN
```

```
C  ACCORDINGLY.  IF THE ORDER IS TO BE INCREASED, WE COMPUTE ONE
C  ADDITIONAL SCALED DERIVATIVE.
C-----------------------------------------------------------------------
 520  PR3 = 1.E+20
      IF (L .EQ. LMAX) GO TO 540
      D1 = 0.
      DO 530 I = 1,N
        D1 = D1 + ((ERROR(I) - Y(I,LMAX))/YMAX(I))**2
      ENQ3 = .5/FLOAT(L+1)
      PR3 = ((D1/EUP)**ENQ3)*1.4 + 1.4E-6
 540  ENQ2 = .5/FLOAT(L)
      PR2 = ((D/E)**ENQ2)*1.2 + 1.2E-6
      PR1 = 1.E+20
      IF (NQ .EQ. 1) GO TO 560
      D = 0.
      DO 550 I = 1,N
 550    D = D + (Y(I,L)/YMAX(I))**2
      ENQ1 = .5/FLOAT(NQ)
      PR1 = ((D/EDN)**ENQ1)*1.3 + 1.3E-6
 560  IF (PR2 .LE. PR3) GO TO 570
      IF (PR3 .LT. PR1) GO TO 590
      GO TO 580
 570  IF (PR2 .GT. PR1) GO TO 580
      NEWQ = NQ
      RH = 1./PR2
      GO TO 620
 580  NEWQ = NQ - 1
      RH = 1./PR1
      GO TO 620
 590  NEWQ = L
      RH = 1./PR3
      IF (RH .LT. 1.1) GO TO 610
      DO 600 I = 1,N
 600    Y(I,NEWQ+1) = ERROR(I)*EL(L)/FLOAT(L)
      GO TO 630
 610  IDOUB = 10
      GO TO 700
 620  IF ((KFLAG .EQ. 0) .AND. (RH .LT. 1.1)) GO TO 610
C-----------------------------------------------------------------------
C  IF THERE IS A CHANGE OF ORDER, RESET NQ, L, AND THE COEFFICIENTS.
C  IN ANY CASE H IS RESET ACCORDING TO RH AND THE Y ARRAY IS RESCALED.
C  THEN EXIT FROM 690 IF THE STEP WAS OK, OR REDO THE STEP OTHERWISE.
C-----------------------------------------------------------------------
      IF (NEWQ .EQ. NQ) GO TO 170
 630  NQ = NEWQ
      L = NQ + 1
      IRET = 2
      GO TO 130
C-----------------------------------------------------------------------
C  CONTROL REACHES THIS SECTION IF 3 OR MORE FAILURES HAVE OCCURED.
C  IT IS ASSUMED THAT THE DERIVATIVES THAT HAVE ACCUMULATED IN THE
C  Y ARRAY HAVE ERRORS OF THE WRONG ORDER.  HENCE THE FIRST
C  DERIVATIVE IS RECOMPUTED, AND THE ORDER IS SET TO 1.  THEN
C  H IS REDUCED BY A FACTOR OF 10, AND THE STEP IS RETRIED.
C  AFTER A TOTAL OF 7 FAILURES, AN EXIT IS TAKEN WITH KFLAG = -2.
C-----------------------------------------------------------------------
 640  IF (KFLAG .EQ. -7) GO TO 670
      RH = .1
      RH = AMAX1(HMIN/ABS(H),RH)
      H = H*RH
      CALL DYN2 (N, T, Y, SAVE1)
      NFE = NFE + 1
      DO 650 I = 1,N
 650    Y(I,2) = H*SAVE1(I)
      IWEVAL = MITER
      IDOUB = 10
      IF (NQ .EQ. 1) GO TO 200
      NQ = 1
      L = 2
      IRET = 3
      GO TO 130
C-----------------------------------------------------------------------
```

```
C  ALL RETURNS ARE MADE THROUGH THIS SECTION.  H IS SAVED IN HOLD
C  TO ALLOW THE CALLER TO CHANGE H ON THE NEXT STEP.
C-----------------------------------------------------------------------
 660  KFLAG = -1
      GO TO 700
 670  KFLAG = -2
      GO TO 700
 680  KFLAG = -3
      GO TO 700
 690  RMAX = 10.
 700  HOLD = H
      JSTART = NQ
      RETURN
C------------------------ END OF SUBROUTINE STIFF ------------------------
      END
C
C
C
      SUBROUTINE PSET (Y, N0, CON, MITER, IER)
C-----------------------------------------------------------------------
C  THE FOLLOWING CARD IS FOR OPTIMIZED COMPILATION UNDER CHAT.
      OPTIMIZE
C-----------------------------------------------------------------------
      DIMENSION Y(N0,1)
      LCM (GEAR6)
      COMMON /GEAR1/ T,H,DUMMY(3),UROUND,N,IDUMMY(3)
      COMMON /GEAR2/ YMAX(1)
      COMMON /GEAR4/ SAVE1(1)
      COMMON /GEAR5/ SAVE2(1)
      COMMON /GEAR6/ PW(1)
      COMMON /GEAR7/ IPIV(1)
      COMMON /GEAR8/ EPSJ,NSQ
C-----------------------------------------------------------------------
C  PSET IS CALLED BY STIFF TO COMPUTE AND PROCESS THE MATRIX
C  P = I - H*EL(1)*J , WHERE J IS AN APPROXIMATION TO THE JACOBIAN.
C  J IS COMPUTED, EITHER BY THE USER-SUPPLIED ROUTINE PEDERV
C  IF MITER = 1, OR BY FINITE DIFFERENCING IF MITER = 2.
C  J IS STORED IN PW AND REPLACED BY P, USING CON = -H*EL(1).
C  THEN P IS SUBJECTED TO LU DECOMPOSITION IN PREPARATION FOR
C  LATER SOLUTION OF LINEAR SYSTEMS WITH P AS COEFFICIENT MATRIX.
C
C  IN ADDITION TO VARIABLES DESCRIBED PREVIOUSLY, COMMUNICATION
C  WITH PSET USES THE FOLLOWING..
C    EPSJ     = SQRT(UROUND), USED IN THE NUMERICAL JACOBIAN INCREMENTS.
C    NSQ      = N0**2.
C-----------------------------------------------------------------------
      IF (MITER .EQ. 2) GO TO 20
C  IF MITER = 1, CALL PEDERV AND MULTIPLY BY SCALAR.--------------------
      CALL PEDERV (N, T, Y, PW, N0)
      DO 10 I = 1,NSQ
 10     PW(I) = PW(I)*CON
      GO TO 60
C  IF MITER = 2, MAKE N CALLS TO DIFFUN TO APPROXIMATE J.---------------
 20   D = 0.
      DO 30 I = 1,N
 30     D = D + SAVE2(I)**2
      R0 = ABS(H)*SQRT(D)*1.E03*UROUND
      J1 = 0
      DO 50 J = 1,N
        YJ = Y(J,1)
        R = EPSJ*YMAX(J)
        R = AMAX1(R,R0)
        Y(J,1) = Y(J,1) + R
        D = CON/R
      CALL DYN2(N,T,Y,SAVE1)
        DO 40 I = 1,N
 40       PW(I+J1) = (SAVE1(I) - SAVE2(I))*D
        Y(J,1) = YJ
        J1 = J1 + N0
 50     CONTINUE
C  ADD IDENTITY MATRIX,-------------------------------------------------
 60   J = 1
      DO 70 I = 1,N
```

```
2914          PW(J) = PW(J) + 1.
2915   70     J = J + (N0 + 1)
2916 C  DO LU DECOMPOSITION ON P.-------------------------------------------------
2917          CALL DEC (N, N0, PW, IPIV, IER)
2918          RETURN
2919 C---------------------- END OF SUBROUTINE PSET ------------------------
2920          END
2921          SUBROUTINE INTERP (TOUT, Y, N0, Y0)
2922 C
2923 CC+-------------------------------------------------------------------------+CC
2924 CC+                                                                         +CC
2925 CC+                        LAWRENCE LIVERMORE LABORATORY                    +CC
2926 CC+       NUMERICAL MATHEMATICS GROUP -- MATHEMATICAL SOFTWARE LIBRARY      +CC
2927 CC+                                                                         +CC
2928 CC+-------------------------------------------------------------------------+CC
2929 CC+                                                                         +CC
2930 CC+                  CLASS ONE ROUTINE:  INTERP                             +CC
2931 CC+                           REVISION:  0                                  +CC
2932 CC+                  DATE LAST CHANGED:  76-02-10                           +CC
2933 CC+                     RELEASE STATUS:  UNLIMITED                          +CC
2934 CC+                                                                         +CC
2935 CC+   EACH  CLASS ONE  ROUTINE HAS BEEN THOROUGHLY TESTED BY NMG AND MEETS  +CC
2936 CC+   CERTAIN DOCUMENTATION AND PROGRAMMING STANDARDS.                       +CC
2937 CC+                                                                         +CC
2938 CC+   AT LEAST ONE CONSULTANT IS AVAILABLE TO ANSWER QUESTIONS AND RESPOND   +CC
2939 CC+   TO REPORTED ERRORS OR INADEQUACIES IN A  CLASS ONE  ROUTINE.           +CC
2940 CC+                                                                         +CC
2941 CC+   +-------------------------------------------------------------+       +CC
2942 CC+   +                           N O T I C E                       +       +CC
2943 CC+   +                                                             +       +CC
2944 CC+   +  THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE +   +CC
2945 CC+   +  UNITED STATES GOVERNMENT.  NEITHER THE UNITED STATES NOR THE  +   +CC
2946 CC+   +  UNITED STATES ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION, +   +CC
2947 CC+   +  NOR ANY OF THEIR EMPLOYEES, NOR ANY OF THEIR CONTRACTORS, SUB- +   +CC
2948 CC+   +  CONTRACTORS, OR THEIR EMPLOYEES, MAKES ANY WARRENTY, EXPRESS OR +   +CC
2949 CC+   +  IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR  +   +CC
2950 CC+   +  THE ACCURACY, COMPLETENESS OR USEFULNESS OF ANY INFORMATION,   +   +CC
2951 CC+   +  APPARATUS, PRODUCT OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS +   +CC
2952 CC+   +  USE WOULD NOT INFRINGE PRIVATELY-OWNED RIGHTS.                 +   +CC
2953 CC+   +                                                               +   +CC
2954 CC+   +-------------------------------------------------------------+       +CC
2955 CC+                                                                         +CC
2956 CC+   PLEASE REPORT ANY SUSPECTED ERRORS IN THIS ROUTINE IMMEDIATELY TO NMG. +CC
2957 CC+                                                                         +CC
2958 CC+-------------------------------------------------------------------------+CC
2959 C
2960 C
2961 C-----------------------------------------------------------------------
2962 C  THE FOLLOWING CARD IS FOR OPTIMIZED COMPILATION UNDER CHAT.
2963          OPTIMIZE
2964 C-----------------------------------------------------------------------
2965          COMMON /GEAR1/ T,H,DUMMY(4),N,IDUMMY(2),JSTART
2966          DIMENSION Y0(N0),Y(N0,1)
2967 C-----------------------------------------------------------------------
2968 C  SUBROUTINE INTERP COMPUTES INTERPOLATED VALUES OF THE DEPENDENT
2969 C  VARIABLE Y AND STORES THEM IN Y0.  THE INTERPOLATION IS TO THE
2970 C  POINT T = TOUT, AND USES THE NORDSIECK HISTORY ARRAY Y, AS FOLLOWS..
2971 C                                   NQ
2972 C                       Y0(I)  =  SUM  Y(I,J+1)*S**J .
2973 C                                 J=0
2974 C  WHERE S = -(T-TOUT)/H.
2975 C-----------------------------------------------------------------------
2976          DO 10 I = 1,N
2977   10       Y0(I) = Y(I,1)
2978          L = JSTART + 1
2979          S = (TOUT - T)/H
2980          S1 = 1.
2981          DO 30 J = 2,L
2982            S1 = S1*S
2983            DO 20 I = 1,N
2984   20         Y0(I) = Y0(I) + S1*Y(I,J)
2985   30       CONTINUE
2986          RETURN
```

```
2987 C---------------------- END OF SUBROUTINE INTERP ----------------------
2988       END
2989       SUBROUTINE COSET (METH, NQ, EL, TQ, MAXDER)
2990 C
2991 C-----------------------------------------------------------------------
2992 C  COSET IS CALLED BY THE INTEGRATOR AND SETS COEFFICIENTS USED THERE.
2993 C  THE VECTOR EL, OF LENGTH NQ + 1, DETERMINES THE BASIC METHOD.
2994 C  THE VECTOR TQ, OF LENGTH 4, IS INVOLVED IN ADJUSTING THE STEP SIZE
2995 C  IN RELATION TO TRUNCATION ERROR.  ITS VALUES ARE GIVEN BY THE
2996 C  PERTST ARRAY.
2997 C  THE VECTORS EL AND TQ DEPEND ON METH AND NQ.
2998 C  COSET ALSO SETS MAXDER, THE MAXIMUM ORDER OF THE METHOD AVAILABLE.
2999 C  CURRENTLY IT IS 12 FOR THE ADAMS METHODS AND 5 FOR THE BDF METHODS.
3000 C  LMAX = MAXDER + 1 IS THE NUMBER OF COLUMNS IN THE Y ARRAY.
3001 C  THE MAXIMUM ORDER USED MAY BE REDUCED SIMPLY BY DECREASING THE
3002 C  NUMBERS IN STATEMENTS 1 AND/OR 2 BELOW.
3003 C
3004 C  THE COEFFICIENTS IN PERTST NEED BE GIVEN TO ONLY ABOUT
3005 C  ONE PERCENT ACCURACY.  THE ORDER IN WHICH THE GROUPS APPEAR BELOW
3006 C  IS..  COEFFICIENTS FOR ORDER NQ - 1, COEFFICIENTS FOR ORDER NQ,
3007 C  COEFFICIENTS FOR ORDER NQ + 1.  WITHIN EACH GROUP ARE THE
3008 C  COEFFICIENTS FOR THE ADAMS METHODS, FOLLOWED BY THOSE FOR THE
3009 C  BDF METHODS.
3010 C-----------------------------------------------------------------------
3011 C
3012 CC+-----------------------------------------------------------------------+CC
3013 CC+                                                                       +CC
3014 CC+                      LAWRENCE LIVERMORE LABORATORY                    +CC
3015 CC+       NUMERICAL MATHEMATICS GROUP -- MATHEMATICAL SOFTWARE LIBRARY     +CC
3016 CC+                                                                       +CC
3017 CC+-----------------------------------------------------------------------+CC
3018 CC+                                                                       +CC
3019 CC+                  CLASS ONE ROUTINE:  COSET                            +CC
3020 CC+                           REVISION:  0                                +CC
3021 CC+                  DATE LAST CHANGED:  76-02-18                         +CC
3022 CC+                     RELEASE STATUS:  UNLIMITED                        +CC
3023 CC+                                                                       +CC
3024 CC+   EACH  CLASS ONE  ROUTINE HAS BEEN THOROUGHLY TESTED BY NMG AND MEETS +CC
3025 CC+   CERTAIN DOCUMENTATION AND PROGRAMMING STANDARDS.                     +CC
3026 CC+                                                                       +CC
3027 CC+   AT LEAST ONE CONSULTANT IS AVAILABLE TO ANSWER QUESTIONS AND RESPOND +CC
3028 CC+   TO REPORTED ERRORS OR INADEQUACIES IN A  CLASS ONE  ROUTINE.         +CC
3029 CC+                                                                       +CC
3030 CC+   +-------------------------------------------------------------+     +CC
3031 CC+   +                           N O T I C E                       +     +CC
3032 CC+   +                                                             +     +CC
3033 CC+   +  THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE +  +CC
3034 CC+   +  UNITED STATES GOVERNMENT.  NEITHER THE UNITED STATES NOR THE  +  +CC
3035 CC+   +  UNITED STATES ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION,  +  +CC
3036 CC+   +  NOR ANY OF THEIR EMPLOYEES, NOR ANY OF THEIR CONTRACTORS, SUB- +  +CC
3037 CC+   +  CONTRACTORS, OR THEIR EMPLOYEES, MAKES ANY WARRENTY, EXPRESS OR +  +CC
3038 CC+   +  IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR  +  +CC
3039 CC+   +  THE ACCURACY, COMPLETENESS OR USEFULNESS OF ANY INFORMATION.   +  +CC
3040 CC+   +  APPARATUS, PRODUCT OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS +  +CC
3041 CC+   +  USE WOULD NOT INFRINGE PRIVATELY-OWNED RIGHTS.                 +  +CC
3042 CC+   +                                                             +     +CC
3043 CC+   +-------------------------------------------------------------+     +CC
3044 CC+                                                                       +CC
3045 CC+   PLEASE REPORT ANY SUSPECTED ERRORS IN THIS ROUTINE IMMEDIATELY TO NMG. +CC
3046 CC+   EXT. 3049, 3329, OR 3288.                                           +CC
3047 CC+                                                                       +CC
3048 CC+-----------------------------------------------------------------------+CC
3049 C
3050       DIMENSION PERTST(12,2,3),EL(13),TQ(4)
3051       DATA  PERTST / 1.,1.,2.,1.,.3158,.07407,.01391,.002182,
3052      |               .0002945,.00003492,.000003692,.0000003524,
3053      2              1.,1.,.5,.1667,.04167,1.,1.,1.,1.,1.,1.,1.,
3054      3              2.,12.,24.,37.89,53.33,70.08,87.97,106.9,
3055      4                126.7,147.4,168.8,191.0,
3056      5              2.0,4.5,7.333,10.42,13.7,1.,1.,1.,1.,1.,1.,1.,
3057      6              12.0,24.0,37.89,53.33,70.08,87.97,106.9,
3058      7                126.7,147.4,168.8,191.0,1.,
3059      8              3.0,6.0,9.167,12.5,1.,1.,1.,1.,1.,1.,1.,1. /
```

```
C
      GO TO (1,2),METH
 1    MAXDER = 12
      GO TO (101,102,103,104,105,106,107,108,109,110,111,112),NQ
 2    MAXDER = 5
      GO TO (201,202,203,204,205),NQ
C-----------------------------------------------------------------------
C  THE FOLLOWING COEFFICIENTS SHOULD BE DEFINED TO MACHINE ACCURACY.
C  FOR A GIVEN ORDER NQ, THEY CAN BE CALCULATED BY USE OF THE
C  GENERATING POLYNOMIAL L(T), WHOSE COEFFICIENTS ARE EL(I)..
C        L(T) = EL(1) + EL(2)*T + ... + EL(NQ+1)*T**NQ.
C  FOR THE IMPLICIT ADAMS METHODS, L(T) IS GIVEN BY
C        DL/DT = (T+1)*(T+2)* ... *(T+NQ-1)/K,     L(-1) = 0,
C  WHERE                  K = FACTORIAL(NQ-1).
C  FOR THE BDF METHODS,
C        L(T) = (T+1)*(T+2)* ... *(T+NQ)/K,
C  WHERE         K = FACTORIAL(NQ)*(1 + 1/2 + ... + 1/NQ).
C
C  THE ORDER IN WHICH THE GROUPS APPEAR BELOW IS..
C  IMPLICIT ADAMS METHODS OF ORDERS 1 TO 12,
C  BDF METHODS OF ORDERS 1 TO 5.
C-----------------------------------------------------------------------
 101  EL(1) = 1.0
      GO TO 900
 102  EL(1) = 0.5
      EL(3) = 0.5
      GO TO 900
 103  EL(1) = 4.16666666666667E-01
      EL(3) = 0.75
      EL(4) = 1.66666666666667E-01
      GO TO 900
 104  EL(1) = 0.375
      EL(3) = 9.16666666666667E-01
      EL(4) = 3.33333333333333E-01
      EL(5) = 4.16666666666667E-02
      GO TO 900
 105  EL(1) = 3.48611111111111E-01
      EL(3) = 1.04166666666667
      EL(4) = 4.86111111111111E-01
      EL(5) = 1.04166666666667E-01
      EL(6) = 8.33333333333333E-03
      GO TO 900
 106  EL(1) = 3.29861111111111E-01
      EL(3) = 1.14166666666667
      EL(4) = 0.625
      EL(5) = 1.77083333333333E-01
      EL(6) = 0.025
      EL(7) = 1.38888888888889E-03
      GO TO 900
 107  EL(1) = 3.15591931216931E-01
      EL(3) = 1.225
      EL(4) = 7.51851851851852E-01
      EL(5) = 2.55208333333333E-01
      EL(6) = 4.86111111111111E-02
      EL(7) = 4.86111111111111E-03
      EL(8) = 1.98412698412698E-04
      GO TO 900
 108  EL(1) = 3.02245370370481E-01
      EL(3) = 1.29642857142857
      EL(4) = 8.68518518518519E-01
      EL(5) = 3.35763888888889E-01
      EL(6) = 7.77777777777778E-02
      EL(7) = 1.06481481481481E-02
      EL(8) = 7.93650793650794E-04
      EL(9) = 2.48015873015873E-05
      GO TO 900
 109  EL(1) = 2.94868000440892E-01
      EL(3) = 1.35892857142857
      EL(4) = 9.76554232804233E-01
      EL(5) = 0.4171875
      EL(6) = 1.11354166666667E-01
      EL(7) = 0.01875
      EL(8) = 1.93452380952381E-03
      EL(9) = 1.11607142857143E-04
```

```
3134          EL(10)= 2.7557319223986E-06
3135          GO TO 900
3136   110    EL(1) = 2.86975446428S7E-01
3137          EL(3) = 1.4144841269841
3138          EL(4) = 1.0772156084656
3139          EL(5) = 4.9856701940035E-01
3140          EL(6) = 0.1484375
3141          EL(7) = 2.9060570987654E-02
3142          EL(8) = 3.7202380952381E-03
3143          EL(9) = 2.9960584656085E-04
3144          EL(10)= 1.3778659611993E-05
3145          EL(11)= 2.7557319223986E-07
3146          GO TO 900
3147   111    EL(1) = 2.8018959644394E-01
3148          EL(3) = 1.4644841269841
3149          EL(4) = 1.1715145502646
3150          EL(5) = 5.7935819003527E-01
3151          EL(6) = 1.8832286155203E-01
3152          EL(7) = 4.1430362654321E-02
3153          EL(8) = 6.2111441798942E-03
3154          EL(9) = 6.2520667989418E-04
3155          EL(10)= 4.0417401528513E-05
3156          EL(11)= 1.5156525573192E-06
3157          EL(12)= 2.5052108385442E-08
3158          GO TO 900
3159   112    EL(1) = 2.7426554003160E-01
3160          EL(3) = 1.5099386724387
3161          EL(4) = 1.2602711640212
3162          EL(5) = 6.5923418209877E-01
3163          EL(6) = 2.3045800264550E-01
3164          EL(7) = 5.5697246105232E-02
3165          EL(8) = 9.4394841269841E-03
3166          EL(9) = 1.1192749669312E-03
3167          EL(10)= 9.0939153439153E-05
3168          EL(11)= 4.8225308641975E-06
3169          EL(12)= 1.5031265031265E-07
3170          EL(13)= 2.8876756987868E-09
3171          GO TO 900
3172   201    EL(1) = 1.0
3173          GO TO 900
3174   202    EL(1) = 6.6666666666667E-01
3175          EL(3) = 3.3333333333333E-01
3176          GO TO 900
3177   203    EL(1) = 5.4545454545455E-01
3178          EL(3) = EL(1)
3179          EL(4) = 9.090909090909E-02
3180          GO TO 900
3181   204    EL(1) = 0.48
3182          EL(3) = 0.7
3183          EL(4) = 0.2
3184          EL(5) = 0.02
3185          GO TO 900
3186   205    EL(1) = 4.3795620437956E-01
3187          EL(3) = 8.2116788321168E-01
3188          EL(4) = 3.1021897810219E-01
3189          EL(5) = 5.4744525547445E-02
3190          EL(6) = 3.6496350364964E-03
3191 C
3192   900    DO 910 K = 1,3
3193   910      TQ(K) = PERTST(NQ,METH,K)
3194          TQ(4) = .5*TQ(2)/FLOAT(NQ+2)
3195          RETURN
3196 C------------------------ END OF SUBROUTINE COSET ------------------------
3197          END
3198          SUBROUTINE DEC (N, NDIM, A, IP, IER)
3199 C-----------------------------------------------------------------------
3200 C  THE FOLLOWING CARD IS FOR OPTIMIZED COMPILATION UNDER CHAT.
3201          OPTIMIZE
3202 C-----------------------------------------------------------------------
3203          LCM (A)
3204          INTEGER N, NDIM, IP(N), IER
3205          REAL A(NDIM,N)
3206 C-----------------------------------------------------------------------
```

```
C  MATRIX TRIANGULARIZATION BY GAUSS ELIMINATION WITH PARTIAL PIVOTING.
C  INPUT..
C     N = ORDER OF MATRIX.
C     NDIM = DECLARED FIRST DIMENSION OF ARRAY  A.
C     A = MATRIX TO BE TRIANGULARIZED.
C  OUTPUT..
C     A(I,J), I.LE.J = UPPER TRIANGULAR FACTOR, U .
C     A(I,J), I.GT.J = MULTIPLIERS = LOWER TRIANGULAR FACTOR, I - L.
C     IP(K), K.LT.N = INDEX OF K-TH PIVOT ROW.
C     IER = 0 IF MATRIX A IS NONSINGULAR, OR K IF FOUND TO BE
C           SINGULAR AT STAGE K.
C  ROW INTERCHANGES ARE FINISHED IN U, ONLY PARTLY IN L.
C  USE  SOL  TO OBTAIN SOLUTION OF LINEAR SYSTEM.
C  IF IER .NE. 0, A IS SINGULAR, SOL WILL DIVIDE BY ZERO.
C-----------------------------------------------------------------------
C
CC+-----------------------------------------------------------------------+CC
CC+                                                                       +CC
CC+                      LAWRENCE LIVERMORE LABORATORY                    +CC
CC+        NUMERICAL MATHEMATICS GROUP -- MATHEMATICAL SOFTWARE LIBRARY   +CC
CC+                                                                       +CC
CC+-----------------------------------------------------------------------+CC
CC+                                                                       +CC
CC+                  CLASS ONE ROUTINE:  DEC                              +CC
CC+                           REVISION:  1                                +CC
CC+                  DATE LAST CHANGED:  76-11-15                         +CC
CC+                     RELEASE STATUS:  UNLIMITED                        +CC
CC+                                                                       +CC
CC+   EACH  CLASS ONE  ROUTINE HAS BEEN THOROUGHLY TESTED BY NMG AND MEETS +CC
CC+   CERTAIN DOCUMENTATION AND PROGRAMMING STANDARDS.                     +CC
CC+                                                                       +CC
CC+   AT LEAST ONE CONSULTANT IS AVAILABLE TO ANSWER QUESTIONS AND RESPOND +CC
CC+   TO REPORTED ERRORS OR INADEQUACIES IN A  CLASS ONE  ROUTINE.        +CC
CC+                                                                       +CC
CC+   +-------------------------------------------------------------+     +CC
CC+   +                          N O T I C E                        +     +CC
CC+   +                                                             +     +CC
CC+   +  THIS REPORT WAS PREPARED AS AN ACCOUNT OF WORK SPONSORED BY THE +  +CC
CC+   +  UNITED STATES GOVERNMENT.  NEITHER THE UNITED STATES NOR THE   +  +CC
CC+   +  UNITED STATES ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION,  +  +CC
CC+   +  NOR ANY OF THEIR EMPLOYEES, NOR ANY OF THEIR CONTRACTORS, SUB- +  +CC
CC+   +  CONTRACTORS, OR THEIR EMPLOYEES, MAKES ANY WARRENTY, EXPRESS OR +  +CC
CC+   +  IMPLIED, OR ASSUMES ANY LEGAL LIABILITY OR RESPONSIBILITY FOR  +  +CC
CC+   +  THE ACCURACY, COMPLETENESS OR USEFULNESS OF ANY INFORMATION,   +  +CC
CC+   +  APPARATUS, PRODUCT OR PROCESS DISCLOSED, OR REPRESENTS THAT ITS +  +CC
CC+   +  USE WOULD NOT INFRINGE PRIVATELY-OWNED RIGHTS.                 +  +CC
CC+   +                                                             +     +CC
CC+   +-------------------------------------------------------------+     +CC
CC+                                                                       +CC
CC+   PLEASE REPORT ANY SUSPECTED ERRORS IN THIS ROUTINE IMMEDIATELY TO NMG. +CC
CC+                                                                       +CC
CC+-----------------------------------------------------------------------+CC
C
      IER = 0
      IF (N .EQ. 1) GO TO 70
      NM1 = N - 1
      DO 60 K = 1,NM1
        KP1 = K + 1
C  FIND THE PIVOT IN COLUMN K.  SEARCH ROWS K TO N. ---------------------
        M = K
        DO 10 I = KP1,N
   10     IF (ABS(A(I,K)) .GT. ABS(A(M,K))) M = I
        IP(K) = M
C  INTERCHANGE ELEMENTS IN ROWS K AND M. --------------------------------
        T = A(M,K)
        IF (M .EQ. K) GO TO 20
        A(M,K) = A(K,K)
        A(K,K) = T
   20   IF (T .EQ. 0.) GO TO 80
C  STORE MULTIPLIERS IN A(I,K), I = K+1,...,N. --------------------------
        T = 1./T
        DO 30 I = KP1,N
   30     A(I,K) = -A(I,K)*T
```

```
C  APPLY MULTIPLIERS TO OTHER COLUMNS OF A. ----------------------------
        DO 50 J = KP1,N
          T = A(M,J)
          A(M,J) = A(K,J)
          A(K,J) = T
          IF (T .EQ. 0.) GO TO 50
          DO 40 I = KP1,N
 40         A(I,J) = A(I,J) + A(I,K)*T
 50       CONTINUE
 60     CONTINUE
 70   K = N
      IF (A(N,N) .EQ. 0.) GO TO 80
      RETURN
 80   IER = K
      RETURN
C---------------------- END OF SUBROUTINE DEC ------------------------
      END
      SUBROUTINE SOL (N, NDIM, A, B, IP)
C-----------------------------------------------------------------------
L  THE FOLLOWING CARD IS FOR OPTIMIZED COMPILATION UNDER CHAT.
      OPTIMIZE
C-----------------------------------------------------------------------
      LCM (A)
      INTEGER N, NDIM, IP(N)
      REAL A(NDIM,N), B(N)
C-----------------------------------------------------------------------
C  SOLUTION OF LINEAR SYSTEM A**X = B USING OUTPUT OF DEC.
C  INPUT..
C     N = ORDER OF MATRIX.
C     NDIM = DECLARED FIRST DIMENSION OF ARRAY  A.
C     A = TRIANGULARIZED MATRIX OBTAINED FROM DEC.
C     B = RIGHT HAND SIDE VECTOR.
C     IP = PIVOT INFORMATION VECTOR OBTAINED FROM DEC.
C  DO NOT USE IF DEC HAS SET IER .NE. 0.
C  OUTPUT..
C     B = SOLUTION VECTOR, X .
C-----------------------------------------------------------------------
      IF (N .EQ. 1) GO TO 50
      NM1 = N - 1
C  APPLY ROW PERMUTATIONS AND MULTIPLIERS TO B. ----------------------
      DO 20 K = 1,NM1
        KP1 = K + 1
        M = IP(K)
        T = B(M)
        B(M) = B(K)
        B(K) = T
        DO 10 I = KP1,N
 10       B(I) = B(I) + A(I,K)*T
 20     CONTINUE
C  BACK SOLVE. -------------------------------------------------------
      DO 40 KB = 1,NM1
        KM1 = N - KB
        K = KM1 + 1
        B(K) = B(K)/A(K,K)
        T = -B(K)
        DO 30 I = 1,KM1
 30       B(I) = B(I) + A(I,K)*T
 40     CONTINUE
 50   B(1) = B(1)/A(1,1)
      RETURN
C---------------------- END OF SUBROUTINE SOL ------------------------
      END
      SUBROUTINE TYPE3
C
C     SUBROUTINE PIPE
C
C     THIS MODULE REPRESENTS A FIXED OR VARIABLE TIME DELAY
C
C     EQUIPMENT PARAMETERS
C     1 - TLAG - LENGTH OF TIME DELAY, IF TIME DELAY IS FIXED
C              - NEGATIVE OF VOLUME OF DELAYING EQUIPMENT OR PIPELINE,
C                  IF TIME DELAY IS VARIABLE
C     2 - BYP  - FRACTION OF STREAM NOT DELAYED, I.E., BYPASSED
```

```
C      3 - NV   - NUMBER OF STORAGE SPACES USED IN DELAY VECTOR
C      4 - FLAG - FLAG.GT.0:OUTPUT FLOW CONTROL (PUMP)
C                 FLAG.EQ.0:NORMAL DELAY
C
       COMMON /UNIT/ IM,NMP
       COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
       COMMON /CON/ NCOMP,NCS
       COMMON /GERR/ JSTART,IMETH,TIME,H,HH,HINC,EPS,T,INTFL
       COMMON /IO/ NIN,NOUT,NERR,NPOINT
       INTEGER OUT
       DIMENSION MC(4), SX(50,1,11)
       DATA MC/4*1/,IFIRST/0/,ND/1/
C
C      TRACE OPTION
       IF (EP(IM,10).GT.0.0) WRITE (NOUT,210) IM
C
       IF (INTFL.EQ.1) GO TO 2
       RETURN
     2 CONTINUE
       IN=MP(IM,3)
       OUT=-MP(IM,4)
       IF (EP(IM,1).LT.0.0) GO TO 10
C
C      FIXED TIME DELAY
       TLAG=EP(IM,1)
       GO TO 20
C
C      VARIABLE TIME DELAY
    10 CONTINUE
       TLAG=-EP(IM,1)/(S(1,IN,3))
 20    CONTINUE
       NVP=EP(IM,2)
       NV=EP(IM,3)
C
C      FLAG=1,OUTPUT FLOW CONTROL ,0 , NORMAL
       IF (EP(IM,4).GT.0.0) S(1,IN,3)=S(1,OUT,3)
C
C      SET UP THE SX MATRIX ON THE FIRST PREDICTOR STEP
C      FOR ALL SUBSEQUENT PREDICTOR STEPS SKIP THE DELAY
       IF (IFIRST.NE.0) GO TO 50
       SX(1,ND,1)=0.0
       SX(1,ND,2)=0.0
C
C      PUT INPUT VALUES IN FIRST VECTOR OF SX MATRIX
       DO 40 K=3,NCS
       SX(1,ND,K)=S(1,IN,K)
C
C      PUT OUTPUT STREAM VALUES IN THE REMAINING VECTORS
       DO 30 I=2,NV
       SX(I,ND,K)=S(1,OUT,K)
C
C      FILL TIME VECTOR INITIALLY WITH -1.0 VALUES
       SX(I,ND,1)=-1.0
       SX(I,ND,2)=-1.0
 30    CONTINUE
 40    CONTINUE
       IFIRST=1
C      RETURN IF PREDICTOR STEP
       RETURN
 50    MC(ND)=MC(ND)+1
C
C      INCREMENT TIME VALUES AS TIME INCREASES
       MD=MC(ND)
       IF (MD.GT.NV) MD=NV
C
C      STORE A COPY OF TIME VECTOR FOR FUTURE REPEATS
       DO 80 I=2,NV
       SX(I,ND,2)=SX(I,ND,1)
 80    CONTINUE
 90    IF (MD.EQ.2) GO TO 100
       SX(MD,ND,1)=SX(MD-1,ND,1)+H
       MD=MD-1
       GO TO 90
```

```
      SX(2,ND,1)=HH
C
C     SHIFT ALL VALUES ONE STORAGE STARTING WITH THE OLDEST
      DO 120 K=3,NC5
      DO 110 I=2,NV
      L=NV-I+2
      SX(L,ND,K)=SX(L-1,ND,K)
110   CONTINUE
120   CONTINUE
C
C     TRANSFER INPUT VALUES TO 1ST DELAY VECTOR
      DO 140 K=3,NC5
      SX(1,ND,K)=S(1,IN,K)
140   CONTINUE
C
C     COMPARE STORED TIME VALUES WITH TIME LAG
      DO 150 I=2,NV
      IF (SX(I,ND,1)-TLAG) 150,160,190
150   CONTINUE
C
C     IF NONE IS GREATER THAN TLAG,EXIT THE LAST VALUE
      I=NV
      IF (SX(NV,ND,1).EQ.-1.) GO TO 160
      WRITE (NOUT,220) IM
      STOP
C
C     IF TIME VALUE EQUALS TLAG,EXIT CORRESPONDING VALUES
160   DO 170 K=3,NC5
      S(1,OUT,K)=SX(I,ND,K)*(1.-BYP)+(S(1,IN,K)*BYP)
170   CONTINUE
      RETURN
C
C     IF SX(I).GT.TLAG,CHECK IF SX(I-1).GT.TLAG
C     IF SO SET THE TIME AT SX(I) TO -1.
180   SX(I,ND,1)=-1.
      I=I-1
190   L=I-1
C
C     KEEP TESTING UNTIL ONLY ONE IS LEFT
      IF (SX(L,ND,1).GE.TLAG) GO TO 180
C
C     INTERPOLATE FOR EXIT VALUE
      A=TLAG-SX(L,ND,1)
      B=SX(I,ND,1)-SX(L,ND,1)
      DO 200 K=3,NC5
      U=SX(L,ND,K)+(A*(SX(I,ND,K)-SX(L,ND,K))/B)
      S(1,OUT,K)=(1.-BYP)*U+(BYP*S(1,IN,K))
200   CONTINUE
      IF (EP(IM,10).GT.0.0) WRITE (NOUT,230)
      RETURN
C
C
210   FORMAT (14H ENTERING DLAY,I3)
220   FORMAT (20H ERROR IN TIME DELAY,/,29H NV MUST BE INCREASED IN MODU
     1,2HLE,I3)
230   FORMAT (19H LEAVING TIME DELAY)
      END
      SUBROUTINE TYPE5
      SUBROUTINE EXTRTR
C
C      PLUTONIUM-URANIUM SEPARATION AND CO-EXTRACTION
C     THE MAIN PROGRAM CONSISTS OF INITIALIZATION FUNCTIONS
C     GENERAL MATERIAL BALANCES, REACTION MODELS, DISTRIBUTION
C     COEFFICIENTS, VARIABLE HOLDUPS, AND CONTROL ALGORITHMS
C     ARE WRITTEN IN SUBROUTINES.
C
C     THE ORGANIC SOLVENT IS ASSUMED TO BE PURE HYDROCARBON WITH
C     TBP OF FRACTION XTBP
C
C     THE AQUEOUS SOLUTION AND FEED STREAMS MAY CONTAIN ALL COMPONENTS
C     THE ORGENIC FEED STREAM MAY CONTAIN PU(IV), U(VI), AND HNO2
C
C     THE PROGRAM IS DIMENSIONED FOR A MAXIMUM OF 20 STAGES
```

```
3499 C
3500       COMMON /LARM/ STD,PRCNT,EMC,EMSTD,EMPRCT,IMC
3501       COMMON /UNIT/ IM,NMP
3502       COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
3503       COMMON /GERR/ JSTART,IMETH,TIME,HELL,HH,HINC,EPS,T,INTFL
3504       COMMON /EXT1/ L, TOR,TAO,OR1,ORF,OR2,N8,NF,VOL,AO,OR
3505       COMMON /EXT2/ K1,K2,K3,K4,K5
3506       DIMENSION EOAP(21),EOAU(21),EH2(21),P40(21),P4A(21),P4AF(21)
3507      ),P3A(21),UA(21),UAF(21),H(21),HF(21),NO3(21),NO3F(21),HNO2F(21)
3508      ),HNO2(21),HN(21),HNF(21),HZ(21),HZF(21),FIS(21),HU(21),ENP(21),
3509      ,ENU(21),ENH2(21),ENH3(21),EMP(21),EMU(21),EMH2(21),EMH3(21),RN(21)
3510      ),UO(21),P3AF(21),EH3(21),DC(21),HNO2O(21),PUORG(21),PUO(21)
3511       DIMENSION AP(21),OP(21),DDTOR(21),DDTAO(21),DDTHU(21)
3512       REAL K1,K2,K3,K4,K5,KH,KP,KU,IS,NO3S,NO3I,INTERR
3513       REAL NO3,NO3F,NO3O(21)
3514       INTEGER AOIN,ORIN,AOOUT,OROUT
3515       IF (INTFL.EQ.1) GO TO 2
3516       RETURN
3517     2 CONTINUE
3518 C
3519 C      IDENTIFICATION OF STREAM NUMBERS
3520 C
3521       IFEEDO=IABS(MP(IM,3))
3522       IFEEDA=IABS(MP(IM,8))
3523       AOIN=IABS(MP(IM,5))
3524       ORIN=IABS(MP(IM,7))
3525       AOOUT=IABS(MP(IM,6))
3526       OROUT=IABS(MP(IM,4))
3527 C
3528 C
3529 C      IDENTIFICATION OF EQUIPMENT PARAMETERS
3530 C
3531       L=EP(IM,2)
3532       NF=EP(IM,3)
3533       VOL=EP(IM,4)
3534       TAO=EP(IM,5)
3535       XTBP=EP(IM,6)
3536       TOR=TAO
3537 C
3538 C      INITIALIZATION
3539 C
3540       M=L+1
3541       N8=NF+1
3542       NZ=NF-1
3543       P4A(M)=S(1,AOIN,7)/239.8
3544       P3A(M)=S(1,AOIN,13)/239.8
3545       UA(M)=S(1,AOIN,6)/238.0
3546       HZ(M)=S(1,AOIN,10)
3547       HNO2(M)=S(1,AOIN,11)
3548       NO3(M)=S(1,AOIN,8)
3549       H(M)=NO3(M)
3550       AO1=S(1,AOIN,3)/60.0
3551       HN(M)=S(1,AOIN,12)
3552       OR1=S(1,ORIN,3)/60.0
3553       P40Z=S(1,IFEEDO,7)/239.0
3554       UOZ=S(1,IFEEDO,6)/238.0
3555       HNO2Z=S(1,IFEEDO,11)
3556       NO3Z=S(1,IFEEDO,8)
3557       ORF=S(1,IFEEDO,3)/60.0
3558       P4AZ=S(1,IFEEDA,7)/239.0
3559       UAZ=S(1,IFEEDA,6)/238.0
3560       NO3AZ=S(1,IFEEDA,8)
3561       HZAZ=S(1,IFEEDA,10)
3562       HNO2AZ=S(1,IFEEDA,11)
3563       HNAZ=S(1,IFEEDA,12)
3564       P3AZ=S(1,IFEEDA,13)/239.0
3565       AOF=S(1,IFEEDA,3)/60.0
3566       IF (T.GT.HH) GO TO 101
3567       PUBAL=0.0
3568 C
3569 C      DEFINE REACTION RATE CONSTANTS FOR CHEMICAL REACTIONS
3570 C
3571       K1=0.0235
3572       K2=0.68966
```

```
      K3=0.0693
      K4=1.5
      K5=0.234
C
C
C     READ IN INITIAL VALUES OF CONCENTRATION PROFILE, DISTRIBUTION
C
      DO 100 I=1,L
      P4A(I)=0.0001
      P3A(I)=0.0001
      UA(I)=0.0001
      H(I)=0.0001
      NO3(I)=0.0001
      HNO2(I)=0.0001
      HN(I)=0.0001
      HZ(I)=0.0001
      EOAP(I)=0.0001
      EOAU(I)=0.0001
      EH3(I)=0.0001
      EH2(I)=0.0001
  100 CONTINUE
      OR2=OR1+OR1
      AO2=AO1+AOF
      INTERP=0.0
      AQFF=AO1
      AQFB=0.0
      ERRPT=0.0
C
C     INITIALIZE PSEUDO FLOW RATES
C
      DO 5 I=1,N7
      AP(I)=AO2
      OP(I)=OR1
    5 DDTOP(I)=0.0
      DO 10 I=N8,L
      AP(I)=AO1
   10 OP(I)=OR2
      AP(NF)=AO2
      OP(NF)=OR2
C
C     INITIALIZE END CONDITIONS
C
      AP(M)=AO1
      DDTAO(M)=0.0
      DDTORF=0.0
C
C     INITIALIZE ORGANIC FLOW RATES AND HOLDUPS
C
      DO 120 I=1,L
      OR=OR2
      IF(I.LT.NF) OR=OR1
      AO=AO2
      IF(I.GT.NF) AO=AO1
      HU(I)=VOL/(1+OR/AO)
      P40(I)=EOAP(I)*P4A(I)
      DDTHU(I)=0.0
  120 CONTINUE
  101 CONTINUE
C
C     OVERALL PU BALANCE
C
      PUBAL=PUBAL+1.0-(AO2*(P4A(1)+P3A(1)+OR2*P4A(L)*EOAP(L))/
     1(ORF*P4OZ+AQF*(P4AZ+P3AZ)))
      EP(IM,1)=PUBAL*(ORF*P4O2+AQF*(P4AZ+P3AZ))*239.0
C
C     CALL SUBROUTINE FOR DISTRIBUTION COEFFICIENTS
C
      CALL DISTRI(P4A,P3A,HN,H,NO3,HNO2,NZ,UA,FIS,EOAU,EOAP,EH3,EH2
     1,XTBP)
C
C     CALL SUBROUTINE FOR HOLDUPS AND FLOW PARAMETERS
C
      CALL HOLDUP(OP,AP,EOAP,EOAU,EH2,EH3,DDTOR,DDTAO,HU,RN,ENP,ENU,
     1ENH2,ENH3,EMP,EMU,EMH2,EMH3,DDTHU,N7,DDTORF)
```

```
C
C     MATERIAL BALANCE CALCULATIONS FOR EIGHT COMPONENTS
C
      DO 300 I=1,L
      OR=OR2
      IF(I.LT.NF) OR=OR1
      AO=AO2
      IF(I.GT.NF) AO=AO1
C
C     CALL SUBROUTINE FOR REACTION MODELS
C
      CALL REACT(P4A,P3A,HN,H,NO3,HNO2,HZ,FIS,RXN1,RXN2,RXN3,RXN4,
     1RXN5,I)
C
C     EULER INTEGRATION ROUTINE, CALCULATE NEW CONCE .RATION
C     PROFILE AT TIME T=T+DT
C
      IF(I.NE.1) GO TO 250
C
C     CALCULATIONS FOR THE FIRST STAGE
C
      TEM1=RN(1)*(P4A(2)-P4A(1))-EMP(1)*P4A(1)-RXN1+RXN2-RXN3
      TEM2=RN(1)*(UA(2)-UA(1))-EMU(1)*UA(1)
      TEM6=RN(1)*NO3(2)-NO3(1)-EMH3(1)*NO3(1)-0.5*RXN2+OR1*NO3Z/HU(1)-
     )4*EMP(1)*P4A(1)-2*EMU(1)*UA(1)
      TEM7=RN(1)*(HNO2(2)-HNO2(1))-EMH2(1)*HNO2(1)+0.5*RXN2-RXN4-RXN5
      TEM8=RN(1)*(H(2)-H(1))*EMH3(1)*NO3(1)+1.75*RXN1-1.5*RXN2+1.25*RXN3
     )+RXN4+RXN5
      GO TO 270
  250 IF(I.NE.NF) GO TO 260
C
C     CALCULATIONS FOR THE FEED STAGE
C
      TEM1=RN(NF+1)*P4A(NF+1)-RN(NF)*P4A(NF)+ORF*P4OZ/HU(NF)+OR1*
     )EOAP(NF-1)*P4A(NF-1)/HU(NF)-EMP(NF)*P4A(NF)-RXN1+RXN2-RXN3
     )+AOF*P4AZ/HU(NF)
      TEM2=RN(NF+1)*UA(NF+1)-RN(NF)*UA(NF)+ORF*UOZ/HU(NF)+OR1*EOAU(NF-1)
     )*UA(NF-1)/HU(NF)-EMU(NF)*UA(NF)
     )+AOF*UAZ/HU(NF)
      TEM6=RN(NF+1)*NO3(NF+1)-RN(NF)*NO3(NF)+ORF*(NO3Z+4*P4OZ+2*UOZ)/
     )HU(NF)+OR1*(EH3(NF-1)*NO3(NF-1)+4*EOAP(NF-1)*P4A(NF-1)+
     )2*EOAU(NF-1)*UA(NF-1))/HU(NF)-EMH3(NF)*NO3(NF)-4*EMP(NF)*P4A(NF)
     )-2*EMU(NF)*UA(NF)-0.5*RXN2
     )+AOF*(NO3AZ+4*P4AZ+2*UAZ)/HU(NF)
      TEM7=RN(NF+1)*HNO2(NF+1)-RN(NF)*HNO2(NF)+OR1*EH2(NF-1)*HNO2(NF-1)
     )/HU(NF)-EMH2(NF)*HNO2(NF)+0.5*RXN2-RXN4-RXN5+ORF*HNO2Z/HU(NF)
     )+AOF*HNO2AZ/HU(NF)
      TEM8=RN(NF+1)*H(NF+1)-RN(NF)*H(NF)+OR1*EH3(NF-1)*NO3(NF-1)/HU(NF)
     )-EMH3(NF)*NO3(NF)+ORF*NO3Z/HU(NF)+1.75*RXN1+1.25*RXN3
     )-1.5*RXN2+RXN4+RXN5+AOF*NO3AZ/HU(NF)
      GO TO 270
C
C     CALCULATIONS FOR THE OTHER STAGES
C
  260 TEM1=RN(I)*(P4A(I+1)-P4A(I))+OR*EOAP(I-1)*P4A(I-1)/HU(I)-EMP(I)*
     )P4A(I)-RXN1+RXN2-RXN3
      TEM2=RN(I)*(UA(I+1)-UA(I))+OR*EOAU(I-1)*UA(I-1)/HU(I)-EMU(I)*UA(I)
      TEM6=RN(I)*(NO3(I+1)-NO3(I))+OR*(EH3(I-1)*NO3(I-1)+4*EOAP(I-1)*
     )P4A(I-1)+2*EOAU(I-1)*UA(I-1))/HU(I)-EMH3(I)*NO3(I)-4*EMP(I)*P4A(I)
     )-2*EMU(I)*UA(I)-0.5*RXN2
      TEM7=RN(I)*(HNO2(I+1)-HNO2(I))+OR*EH2(I-1)*HNO2(I-1)/HU(I)-EMH2(I)
     )*HNO2(I)+0.5*RXN2-RXN4-RXN5
      TEM8=RN(I)*(H(I+1)-H(I))+OR*EH3(I-1)*NO3(I-1)/HU(I)-EMH3(I)*NO3(I)
     )+1.75*RXN1-1.5*RXN2+1.25*RXN3+RXN4+RXN5
C
C     CALCULATE THE CONCENTRATION PROFILE AT TIME T=T+DT
C
      DT=HH/60.0
  270 P4AF(I)=P4A(I)+(TEM1-(1-EOAP(I))*P4A(I)*DDTHU(I)/HU(I))*DT/(1+
     )ENP(I))
      IF(P4AF(I).LE.0.0) P4AF(I)=0.0
      UAF(I)=UA(I)+(TEM2-(1-EOAU(I))*UA(I)*DDTHU(I)/HU(I))*DT/(1+ENU(I))
      IF(UAF(I).LT.0.0) UAF(I)=0.0
```

```
3720        TEM3=PH(I)*(P3A(I+1)-P3A(I))+RXN1-RXN2+RXN3
3721        P3AF(I)=P3A(I)+(TEM3-P3A(I)*DDTHU(I)/HU(I))*DT
3722        IF(P3AF(I).LT.0.0) P3AF(I)=0.0
3723        TEM4=PH(I)*(HN(I+1)-HN(I))-K2*RXN1-RXN5
3724        HNF(I)=HN(I)+(TEM4-HN(I)*DDTHU(I)/HU(I))*DT
3725        IF(HNF(I).LT.0.0) HNF(I)=0.0
3726        TEM5=PH(I)*(HZ(I+1)-HZ(I))-0.25*RXN3-RXN4
3727        HZF(I)=HZ(I)+(TEM5-HZ(I)*DDTHU(I)/HU(I))*DT
3728        IF(HZF(I).LT.0.0) HZF(I)=0.0
3729        NO3F(I)=NO3(I)-(4*(P4AF(I)-P4A(I))*EHP(I)+2*(UAF(I)-UA(I))*ENU(I)-
3730       1(TEM6-(1-EH3(I))*NO3(I)*DDTHU(I)/HU(I))*DT)/(1+ENH3(I))
3731        IF(NO3F(I).LT.0.0) NO3F(I)=0.0
3732        HNO2F(I)=HNO2(I)+(TEM7-(1-EH2(I))*HNO2(I)*DDTHU(I)/HU(I))*DT/(1+
3733       1ENH2(I))
3734        IF(HNO2F(I).LT.0.0) HNO2F(I)=0.0
3735        HF(I)=H(I)+(TEM8-(H(I)-EH3(I)*NO3(I))*DDTHU(I)/HU(I))*DT-ENH3(I)*
3736       1(NO3F(I)-NO3(I))
3737        IF(HF(I).LT.0.0) HF(I)=0.0
3738    300 CONTINUE
3739 C
3740 C
3741 C       ESTABLISH NEW PRESENT VALUES
3742 C
3743    354 DO 450 I=1,L
3744        AP(I)=AP(I)+DDTAQ(I)*DT
3745        OP(I)=OP(I)+DDTOP(I)*DT
3746        P4A(I)=P4AF(I)
3747        P3A(I)=P3AF(I)
3748        UA(I)=UAF(I)
3749        HZ(I)=HZF(I)
3750        HN(I)=HNF(I)
3751        HNO2(I)=HNO2F(I)
3752        NO3(I)=NO3F(I)
3753        H(I)=HF(I)
3754    450 CONTINUE
3755 C
3756 C
3757 C       CALCULATE CONCENTRATIONS IN ORGANIC PHASE
3758 C
3759    444 DO 470 I=1,L
3760        HNO2O(I)=HNO2(I)*EH2(I)
3761        NO3O(I)=NO3(I)*EH3(I)
3762        UO(I)=UA(I)*EOAU(I)
3763    470 PUO(I)=P4A(I)*EOAP(I)
3764 C
3765 C       COMPUTE OUTPUT VALUES
3766 C
3767        S(1,AOOUT,6)=UA(1)*238.0
3768        S(1,AOOUT,7)=P4A(1)*239.0
3769        S(1,AOOUT,8)=NO3(1)
3770        S(1,AOOUT,10)=HZ(1)
3771        S(1,AOOUT,11)=HNO2(1)
3772        S(1,AOOUT,12)=HN(1)
3773        S(1,AOOUT,13)=P3A(1)*239.0
3774        S(1,OPOUT,6)=UO(L)*238.0
3775        S(1,OPOUT,7)=PUO(L)*239.0
3776        S(1,OPOUT,8)=NO3O(L)
3777        S(1,OPOUT,11)=HNO2O(L)
3778         RETURN
3779         END
3780 C
3781 C
3782 C       LISTING OF SUBROUTINES USED IN EXTRTR
3783 C
3784        SUBROUTINE DISTRI(P4A,P3A,HN,H,NO3,HNO2,HZ,UA,
3785       1FIS,EOAU,EOAP,EH3,EH2,XTBP)
3786 C
3787        COMMON/EXT1/L,TOR,TAQ,OR1,ORF,OR2,NB,NF,VOL,AQ1,OR,AQ2
3788        COMMON /EXT2/ K1,K2,K3,K4,K5
3789        DIMENSION P4A(21),P3A(21),HN(21),H(21),NO3(21),HNO2(21),HZ(21)
3790       1,FIS(21)
3791        DIMENSION UA(21),EOAU(21),EOAP(21),EH3(21),EH2(21)
3792        REAL K1,K2,K3,K4,K5,KH,KP,KU,IS,NO3S,NO3I,INTERR
```

```
      REAL NO3,NO3F,NO3O(21)
      DO 150 I=1,L
      W=NO3(I)**2
      IS=0.5*(NO3(I)+HI(I)+HN(I)+HZ(I))+2*UA(I)+8.*P4A(I)+4.5*P3A(I)
      KP=12.163-9.033*IS+2.23*IS*IS-0.163*IS**3
      KU=8.291+6.071*IS-6.176*IS*IS+1.579*IS**3
      KH=0.385-0.155*IS+0.024*IS*IS
      FIS(I)=10**(0.9*SQRT(IS)-1.521)
      C=(XTBP(0.2)*0.731-EH2(I)*HNO2(I)
      A1=1+KH*H(I)*NO3(I)
      A2=UA(I)*KP*P4A(I)*W/KU
      A3=P4A(I)*KU*UA(I)/(KP*W)
      A4=(A1/NO3(I))**2
      A5=A4*W
      A6=A1+4*SQRT(KU)*NO3(I)*A2)
      A7=A1+4*SQRT(KP)*W*A3)
      A8=1-SQRT(1+8*C*KU*A2/A4)
      A9=1-SQRT(1+8*C*KP*A3/A5)
      EOAU(I)=(A6*A8)**2
      EOAP(I)=(A7*A9)**2
      B1=KH*H(I)+1/NO3(I)
      B2=UA(I)*KU+W*P4A(I)*KP
      B3=1-SQRT(1+8*C*B2/(B1*B1))
      EH3(I)=(B1*KH*B3/(4*B2)
      FTBP=(XTBP(0.2)*0.731-EH3(I)*NO3(I)-2*EOAP(I)*P4A(I)-2*EOAU(I)
     )*UA(I)
      IF(FTBP.LT.0.0) FTBP=0.0
  150 EH2(I)=10.5*FTBP
      RETURN
      END
C
C
      SUBROUTINE HOLDUP(OP,AP,EOAP,EOAU,EH2,EH3,DDTOR,DDTAO,HU,RN,ENP,
     1ENU,ENH2,ENH3,EMP,EMU,EMH2,EMH3,DDTHU,NF,DDTORF)
C
      COMMON /EXT1/ L,TOR,TAO,OR1,ORF,OR2,N8,NF,VOL,AO,OR
      COMMON /EXT2/ K1,K2,K3,K4,K5
      DIMENSION OP(21),AP(21),EOAP(21),EOAU(21),EH2(21),EH3(21),
     1DDTOR(21),DDTAO(21),HU(21)
      DIMENSION RN(21),ENP(21),ENU(21),ENH2(21),ENH3(21),EMP(21),
     2EMU(21),EMH2(21),EMH3(21),DDTHU(21)
      REAL K1,K2,K3,K4,K5,KH,KP,KU,IS,NO3O,NO3I,INTERR
      REAL NO3,NO3F,NO3O(21)
C
C     CALCULATE HOLDUPS AND FLOW PARAMETERS
C
      DDTOR(NF)=(1/TOR)*(OP(NF-1)+ORF-OP(NF))
      DO 160 I=N8,L
  160 DDTOR(I)=(1/TOR)*(OP(I-1)-OP(I))
      DO 200 I=1,L
      OR=OR2
      IF(I.LT.NF) OR=OR1
      AO=AO2
      IF(I.GT.NF) AO=AO1
      DDTAO(I)=(1/TAO)*(AP(I+1)-AP(I))
      HU(I)=VOL/(1+OP(I)/AP(I))
      RN(I)=AO/HU(I)
      ENP(I)=OP(I)*EOAP(I)/AP(I)
      ENU(I)=OP(I)*EOAU(I)/AP(I)
      ENH2(I)=OP(I)*EH2(I)/AP(I)
      ENH3(I)=OP(I)*EH3(I)/AP(I)
      EMP(I)=OR*EOAP(I)/HU(I)
      EMU(I)=OR*EOAU(I)/HU(I)
      EMH2(I)=OR*EH2(I)/HU(I)
  200 EMH3(I)=OR*EH3(I)/HU(I)
      DDTHU(1)=(VOL*OR1*DDTAO(2))/((AP(2)+OR1)**2)
      DO 202 I=2,N7
  202 DDTHU(I)=(VOL*OP(I-1)*DDTAO(I+1))/((AP(I+1)+OP(I-1))**2)
      DDTHU(NF)=(VOL*(OP(N7)+ORF)*DDTAO(N8)-VOL*(AP(N8)+AOF)*DDTORF)/((
     )AP(N8+AOF)+OR1+ORF)**2)
      DO 204 I=N8,L
  204 DDTHU(I)=(VOL*OP(I-1)*DDTAO(I+1)-VOL*AP(I+1)*DDTOR(I-1))/((AP(I+1)
     )+OP(I-1))**2)
      RETURN
      END
```

```
3868 C
3869 C
3870       SUBROUTINE REACT(P4A,P3A,HN,H,NO3,HNO2,HZ,FIS,RXN1,RXN2,RXN3,RXN4
3871      1,RXN5,I)
3872 C
3873       COMMON /EXTI/ L,TDP,TAQ,DP1,DRF,DR2,HB,HF,VOL,AO1,DP,AO2
3874       COMMON /EXT2/ K1,K2,K3,K4,K5
3875        DIMENSION P4A(21),P3A(21),HN(21),H(21),NO3(21),HNO2(21),HZ(21)
3876      1,FIS(21)
3877       REAL K1,K2,K3,K4,K5,KH,KP,KU,IS,NO3S,NO3I,INTERR
3878       REAL NO3,NO3F,NO3O(21)
3879 C     CHEMICAL REACTION MODELS
3880       RXN1=K1*(HN(I)**2)*(P4A(I)**2)/((P3A(I)**2)*(H(I)**4)*((0.19+
3881      1NO3(I)**2)))
3882       RXN2=K4*P3A(I)*HNO2(I)*H(I)*NO3(I)
3883       RXN3=K3*P4A(I)*HZ(I)
3884       RXN4=K5*HZ(I)*HNO2(I)*H(I)*H(I)
3885       RXN5=IS(I)*HNO2(I)
3886       IF (H(I),LE,0.0) RXN5=0.0
3887        RETURN
3888        END
3889       SUBROUTINE TYPE6
3890 C
3891 C     SUBROUTINE PRECIP
3892 C
3893 C     THIS MODULE DESCRIBES THE PLUTONIUM OXALATE PRECIPITATOR
3894 C
3895 C     THE FOLLOWING PARAMETERS ARE REQUIRED
3896 C     EP(IM,1)--PU HOLDUP IN GRAMS
3897 C     EP(IM,2)   VOLUME OF PRECIPITATOR (LITERS)
3898 C
3899       COMMON/MAT/MP(35,13),EP(35,10),S(2,45,13),EX(50)
3900       COMMON/CON/NCOMP,NCS,NE,NS,TMAX
3901       COMMON/UNIT/IM,NMP
3902       COMMON/GERR/JSTART,IMETH,TIME,H,HH,HINC,EPS,T,INTFL
3903       DIMENSION Y(8),DERY(8)
3904 C
3905 C     THE MICROSCOPIC QUANTITIES IN THE DATA STATEMENT HAVE THE UNITS
3906 C     AKG(DM/HR), AKB(L/LITERS*HOUR), R2RO(DM), RO(G/CC)
3907       DATA AKG,ANG,AKB,ANB,R2RO/1.5E-04,1.,1.E+08,1.4,1.E-06/
3908       DATA RO,VF/2.694,4.1887902/
3909       IF (INTFL.EQ.0) GO TO 2
3910       RETURN
3911     2 CONTINUE
3912 C
3913 C     VOLUME OF PRECIPITATOR
3914       V=EP(IM,2)
3915 C
3916       ITER=0
3917 C     FIND STREAMS
3918       INPU=MP(IM,3)
3919       INOX=MP(IM,4)
3920       NOUT=IABS(MP(IM,5))
3921       IF (MP(IM,6),EQ,0) GO TO 4
3922       NSIG=IABS(MP(IM,6))
3923     4 CONTINUE
3924 C
3925 C
3926 C     DENSITY OF PU IN OXALATE (G/LITER)
3927 C     DENSITY OF C2O4 IN OXALATE (MOLES/LITER)
3928 C
3929       ROP=457.*RO
3930       ROO=3.82*RO
3931 C
3932 C
3933 C     CALCULATE INITIAL CONDITIONS
3934 C      FLOW OF FILTRATE, L/HR
3935       Y(2)=S(1,NOUT,6)*60.0
3936 C      CONC OF PU
3937       Y(1)=S(1,NOUT,7)
3938 C      CONC OF NO3
3939       Y(3)=S(1,NOUT,8)
3940 C      CONC OF C2O4
```

```
3941          Y(4)=S(1,NOUT,9)
3942          Y(5)=S(1,NOUT,12)
3943          Y(6)=S(1,NOUT,13)
3944          Y(7)=S(1,NOUT,10)
3945          Y(8)=S(1,NOUT,11)
3946    C
3947      1   CONTINUE
3948    C
3949    C     COMPUTE INPUT VARIABLES
3950      3   IFPDT=S(1,INPU,4)*3600.0
3951          IFODT=S(1,INOX,4)*3600.0
3952          FP=S(1,INPU,3)*60.0
3953          FO=S(1,INOX,3)*60.0
3954          CP=S(1,INPU,7)
3955          VK=S(1,INPU,8)
3956          CO=S(1,INOX,9)
3957          TFP=FO+FP
3958          TDFP=IFODT+IFPDT
3959    C
3960    C     COMPUTE PLUTONIUM EQUILIBRIUM SOLUBILITY
3961          CALL SOLUB(Y(3),Y(4),CSTAR,CH)
3962          CSTAR=CSTAR*239.
3963    C
3964          V=1.
3965          IF ((Y(3)-CSTAR).GT.0.) V=0.
3966    C
3967    C     COMPUTE GROWTH AND NUCLEATION COEFFICIENTS
3968          G=AKG*V**ABS(Y(3)-CSTAR))**ANG
3969          B=AKB*V**ABS(Y(3)-CSTAR))**ANB
3970    C
3971          IF (Y(2).EQ.0.0.OR.TFP.EQ.0.) GO TO 50
3972    C
3973    C     SYSTEM OF DIFFERENTIAL EQUATIONS
3974          P1=3.*Y(2)*VK*V*G*R*RO+V*VK*Y(2)*B*RZRO**3/TFP
3975          P2=(CP*Y(1)+FP-Y(1)*FO+Y(1)-ROP)*P1
3976          DERY(1)=P2*TFP/(V*Y(2))
3977          X1=TFP*V-Y(2)*V*3.*VK*Y(7)*G/RZRO-Y(2)*VK*RZRO**3*B/TFP
3978          X2=Y(2)*TDFP/TFP
3979          DERY(2)=X2+TFP*X1
3980          Z1=-DERY(2)/Y(2)+TDFP/TFP
3981          Z2=(FP*(H/Y(2)-Y(3))*TFP/V
3982          DERY(3)=Z1*Y(3)+Z2
3983          T1=(ROO-Y(4))*TDFP/TFP
3984          T2=TFP*(FO*CO-TFP*ROO-Y(2)*Y(4)+Y(2)*ROO)/(V*Y(2))
3985          T3=(ROO-Y(4))*DERY(2)/Y(2)
3986          DERY(4)=-T1+T2+T3
3987          TRAV=B*Y(2)*RZRO**3/TFP
3988          DERY(5)=TRHV-Y(5)*TFR/V
3989          DERY(6)=G*Y(5)/RZRO+TRAV-Y(6)*TFR/V
3990          DERY(7)=2.*G*Y(6)/RZRO+TRAV-Y(7)*TFR/V
3991          AMU3=(1.-Y(2))/TFR)/VK
3992          DERY(8)=4.*G*AMU3/RZRO+TRAV-Y(8)*TFR/V
3993    C
3994          GO TO 60
3995    C
3996    C     LIMIT OF SYSTEM OF DIFFERENTIAL EQUATIONS
3997    C     FOR VANISHING INPUT FLOW
3998     50   DD=B*RZRO**3
3999          DERY(1)=-ROP*VK*DD
4000          DERY(2)=DFODT+DFPDT
4001          DERY(3)=0.
4002          DERY(4)=0.
4003          DERY(5)=DD
4004          DERY(6)=DD
4005          DERY(7)=DD
4006          DERY(8)=DD
4007     60   CONTINUE
4008    C
4009    C
4010    C     COMPUTE OUTPUT SIGNAL
4011          IF (CP.LE.1.0E-20.OR.FP.LE.1.E-20) GO TO 30
4012          SIGNL=Y(1)*Y(2)/(CP*FP)
4013          GO TO 40
```

```
      30  SIGNL=0.
C
C     COMPUTE OUTPUT VARIABLES
      40 IF (MP(IM,6).EQ.0) GO TO 41
         S(1,NSIG,3)=SIGNL
      41 CONTINUE
         S(1,NOUT,3)=TFP/60.0
         S(2,NOUT,6)=DEPY(2)/3600.0
         S(2,NOUT,7)=DEPY(1)
         S(2,NOUT,8)=DERY(3)
         S(2,NOUT,9)=DEPY(4)
         S(2,NOUT,12)=DERY(5)
         S(2,NOUT,13)=DERY(6)
         S(2,NOUT,10)=DEPY(7)
         S(2,NOUT,11)=DERY(8)
C
C      PU HOLDUP IN THE REACTOR
C
         EP(IM,1)=S(1,NOUT,7)*V
C
         RETURN
         END
         SUBROUTINE SOLUB(CNO3,CHCO,CSTAR,CH)
         DIMENSION X(3)
         G=2.0943951
         B=-CNO3/3.
         C=-5.38E-2*CHCO/3.
         D=-5.58E-6*CHCO
         Q=C-B**2
         R=(3.*B*C-B)/2.-B**3
         DELT=Q**3+R**2
         IF (DELT.LE.0.) GO TO 10
         SD=SQRT(DELT)
         S1=(R+SD)**0.33333333
         S2=(R-SD)**0.33333333
         CH=S1+S2-B
         GO TO 50
      10 S=R/ABS(R)
         T1=2.*S*SQRT(-Q)
         T2=ACOS(S*R/SQRT(-Q**3))/3.
         X(1)=T1*COS(T2)-B
         X(2)=T1*COS(T2+G)-B
         X(3)=T1*COS(T2+2.*G)-B
         DO 100 I=1,3
         IF(X(I).LT.0.) GO TO 100
         CH=X(I)
      100 CONTINUE
      50 A1=4.890873E-05
         B1=6.124839E-11
         C1=4.985781E-04
         CSTAR=A1+B1*CH**4/CHCO**2+C1*CHCO
         RETURN
         END
         SUBROUTINE TYPE7
C
C           SUBROUTINE EVAPTR
C
         COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
         COMMON /CON/ NCOMP,NC5,NE,NS,TMAX,NC3,N8,N1,N2,NF
         COMMON /GERR/ JSTART,IMETH,TIME,H,HH,HINC,EPS,T,INTFL
         COMMON /IO/ NIN,NOUT,NERR,NPOINT,NPRT
         COMMON/UNIT/IM,NMP
         COMMON /PTAB/ TREF,R
         INTEGER OUT,VAP,STEAM
         REAL LMDA,MFIN,MOUT,MSTM
         IF (INTFL.EQ.1) GO TO 1
         RETURN
       1 CONTINUE
C
C           EQUIPMENT PARAMETERS
C
C                1 - PU-HOLDUP,G
C                2 - VOLUME,L
```

```
C
      V=EP(IM,2)*1000.0
C
C           CALLS TO PROPERTY SUBROUTINES DENL AND VAPR
C
C          FIND STREAMS
C
      IN=IABS(MF(IM,3))
      STEAM=IABS(MF(IM,4))
      OUT=IABS(MF(IM,5))
      VAP=IABS(MF(IM,6))
C
C           INPUT VALUES
C
      CALL DENL (1,IN,DAVGIN)
      MFIN=S(1,IN,3)*55.5*60.0*DAVGIN
      XIPU=S(1,IN,2)*239.0/55.5/DAVGIN
      HSTM=S(1,STEAM,3)*18.0/60.0
      CALL VAPR (1,OUT,P,PS,DPV)
      CTIN=70.0
      XOPU=0.0
      XOPL=0.0
      V=0.0013
      CSTM=25.56
C
C           LATENT HEAT OF STEAM AS A FUNCTION OF TEMPERATURE
C
      TMDA=0.36
      HSTM=1075.+TMDA
C
C           LATENT HEAT OF VAPOR LEAVING THE LIQUID IN THE CONCENTRATOR
C
      HVAP=S(1,OUT,7)/239.0/55.5/DAVGOUT
      IF (MOUT.GE.0.0) GO TO 3
      IF (MOUT.LT.0.0) GO TO 2
    2 WRITE (NOUT,100)
  100 FORMAT (5X,"NEGATIVE MOLAR FLOW RATE.  COMPUTATIONS CEASE.")
      CALL EXIT
C
C           SUBROUTINE TO CALCULATE FLUID TEMPERATURE IN MIXER CAN BE
C           ADDED HERE IF NECESSARY.
C
    3 XIW=1.0-XIPU
      P1=(DAVGOUT*V)
      APHA=(MFIN*XIPU)/P1*0.23975
      BETA=MOUT/P1*0.23975
      DLTA=MOUT/P1*0.018016
      GAMA=((MFIN*XIW)-(HSTM/HVAP))/P1*0.018016
      XOW=1.0-XOPU
      X=(APHA-(APHA-BETA*XOPU)*EXP((-1)*BETA*(T-HH)/60.0))/BETA
      XW=(GAMA-(GAMA-DLTA*XOW)*EXP((-1)*DLTA*(T-HH)/60.0))/DLTA
C
C           IN PLACE OF X AND XW THE VALUES OF DX/DT AND D(XW)/DT COULD BE
C           CALCULATED HERE.  VALUES OF S(2,OUT,7) (DERIVATIVES) WOULD THEN
C           BE REPORTED OUT OF THE SUBROUTINE FOR INTEGRATION BY THE
C           INTEGRATOR SUBROUTINE "DRIVE".  THE STREAM LABELED "OUT" WOULD
C           BE A COUPLED STREAM AND THE TEST VALUE FOR INTFL MUST BE
C           CHANGED TO 0.
C
      S(1,OUT,7)=239.0*55.5*DAVGOUT*X/(X+XW)
      S(1,OUT,3)=MOUT/60.0/55.5/DAVGOUT
      S(1,VAP,3)=((MFIN-MOUT)/60.0)*R*TEMP/PRESS
      EP(IM,1)=S(1,OUT,7)*V*1000.0
      RETURN
      END
```

```
      SUBROUTINE TYPE2
C
C         SUBROUTINE CONTLR
C
C         PROPORTIONAL-INTEGRAL (PI) CONTROLLER
C
C         EQUIPMENT PARAMETERS
C
C         1 - CONTROLLED VARIABLE NUMBER
C         2 - RANGE OF CONTROLLED VARIABLE
C         3 - SET POINT
C         4 - PROPORTIONAL GAIN
C         5 - INTEGRAL CONSTANT
C
      COMMON /MAT/ MP(35,13),EP(35,10),S(2,45,13),EX(50)
      COMMON /CON/ NCOMP,NCS,NE,NS,TMAX,NC3,NB,N1,N2
      COMMON /GEPP/ JSTART,IMETH,TIME,H,HINC,EPS
      COMMON /UNIT/ IM,NMP
      COMMON /PTAB/ TPEF,P
      DATA OLD/0.0/,OLDOUT/0.0/
      COMMON /PROP/MW(6),CPL(6,6),CVA(6,6),ENT(6,6),ENV(6,6),LAM(6,6)
     1,VAP(6,3),WI(6,6),DHL(6,6)
      INTEGER OUT
C
      P1=EP(IM,4)
      P2=EP(IM,5)
      IN=MP(IM,3)
      OUT=IABS(MP(IM,4))
      K=EP(IM,1)
C         MEASURED VARIABLE AT CURRENT TIME
      SIG1=S(1,IN,K)
C         MEASURED VARIABLE AT LAST TIME
      SIG2=OLD
      OLD=S(1,IN,K)
      SCALE=1.0/EP(IM,2)
C         PRESENT ERROR
      ERR=(SIG1-EP(IM,3))*SCALE
C         LAST ERROR
      OLDER=(SIG2-EP(IM,3))*SCALE
C         OUTPUT SIGNAL
      S(1,OUT,3)=P1*(ERR-OLDER+P2*(ERR+OLDER)*0.5*HINC)+OLDOUT
      OLDOUT=S(1,OUT,3)
      IF(S(1,OUT,3).GT.100) S(1,OUT,3)=1.0
      IF(S(1,OUT,3).LT.0.0) S(1,OUT,3)=0.0
      RETURN
      END
      SUBROUTINE TYPE9
C
C         SUBROUTINE VALV
C
C         CONTROL VALVE          (LINEAR RESPONSE)
C
C         EQUIPMENT PARAMETERS
C              EP(IM,1)=CONSTANT
C              EP(IM,2)=ACTION(+=DIRECT,-=REVERSE)
C
      COMMON /MAT/MP(35,13),EP(35,10),S(2,45,13),EX(50)
      COMMON/UNIT/ IM,NMP
      IN=MP(IM,3)
      OUT=IABS(MP(IM,4))
      A=EP(IM,2)
      IF(A.LT.0.0) S(1,OUT,3)=EP(IM,1)*(1.0-S(1,IN,3))
      IF(A.GE.0.0) S(1,OUT,3)=EP(IM,1)*S(1,IN,3)
      RETURN
      END
      SUBROUTINE TYPE8
C
C         SUBROUTINE GNTRNS
C
C              A MODULE FOR THE SIMULATION OF HEAT AND MASS TRANSFER IN
C         A SINGLE STAGE WITH EITHER LIQ-LIQ OR VAPOR-LIQ PHASES OR ONE
C         WELL MIXED PHASE
C
```

```
      COMMON  MAT/MP(35,13),EP(35,10),S(2,45,3),EX(50)
      COMMON  CONANCOMP,NCS,NE,NS,TMAX,NC3,NB,NIST,NFIN
      COMMON  GEPP/JSTART,IMETH,TIME,H,HINC,EPS
      COMMON  UNIT,IM,NMP
      REAL    IN,IOUT,IV,KEXT
      INTEGER OUTPPD,OUTSTD,OUTPRA,OUTSTA
      DIMENSION YD(10),XD(10),XSLF(10),XD(13,8),TD(8),X(13,8),W(8),T(8)
      DIMENSION YIN(10),VOL(2),XOUT(8),DERY(10),DERX(10),YIN(10)
      DIMENSION YOUT(10),DISCO(10),YIDEAL(10),P(8),XX(13,8),XXX(13,8)
      DIMENSION TRANS(10),TRANSUM(10)
C
C         EQUIPMENT PARAMETERS
C
C               1-HOLDUP OF COMP 2 (VAR 7)
C               2 TOTAL VOLUME (VOL(1))
C               3 VOLUME OF AQ OR LIQ PHASE (VOL(2))
C               4 RATE CONSTANT FOR VOLUME CHANGE (KV)
C               5 RATE CONSTANT FOR APPR  TO EQUIL (KEXT)
C               6 VOL FRACTION EXTR ENHANCEMENT COMP (XTBP)
C               7 HEAT XFER COEF (HCOEF)
C               8 HEAT XFER AREA (HAREA)
C               9-NTYPE=0 FOR EXTRACTION;1 FOR FLASH(LIQ-VAP)
C
      VOL(1)=EP(M,2)
      VOL(2)=EP(M,3)
      XTBP=EP(M,6)
      KV=EP(M,4)
      KEXT=EP(M,5)
      HCOEF=EP(M,7)
      HAREA=EP(M,8)
      NTYPE=EP(M,9)
C
C         INLAND OUT STREAMS
C
C               LIQ OR AQUEOUS
      INSTGA=IABS(MP(M,7))
      INFDA=IABS(MP(M,8))
C         STREAM OUTPPA IS THE LEVEL CONTROL STREAM
      OUTPRA=IABS(MP(M,9))
      OUTSTA=IABS(MP(M,10))
C               VAP OR ORGANIC
      INSTGD=IABS(MP(M,3))
      INFDD =IABS(MP(M,4))
      OUTPRD=IABS(MP(M,5))
      OUTSTD=IABS(MP(M,6))
      JSIGN=IABS(MP(M,12))
      JENPGV=IABS(MP(M,11))
      JCONT=IABS(MP(M,13))
C
      NED=NCOMP+1
C
C     INITIAL VALUES
      NST=0
      DO 26 I=3,NC5
      DO 25 J=1,NST
      IA=IABS(MP(M,J+2))
      IF(IA.EQ.0) GO TO 24
      XX(I-2,J)=S(1,IA,1)
      GO TO 25
   24 XX(I-2,J)=0.0
   25 CONTINUE
   26 CONTINUE
C
C     RENAME STREAM VARIABLES
   28 CONTINUE
      DO 30 J=1,NST
      W(J)=XX(1,J)
      T(J)=XX(2,J)
      P(J)=XX(3,J)
      DO 29 I=4,NC3
      X(I-3,J)=XX(I,J)
   29 CONTINUE
C     ****      FOR U-PU EXTRACTION
```

```
      X(1,J)=X(1,J)/238.0
      X(2,J)=X(2,J)/239.0
C     ****
   30 CONTINUE
C
C         COMPUTE ENTHALPIES,    HEAT CAPACITIES, DENSITIES, AND HEATS
C              OF VAPORIZATION
      CALL ENTL(1,INSTGA,HTLIN1,DH)
      CALL ENTL(1,INFDA,HTLIN2,DH)
      CALL ENTL(1,OUTSTA,HTLOUT1,DH)
      CALL ENTL(1,OUTPRA,HTLOUT2,DH)
      HTLIN=HTLIN1        *W(5)+HTLIN2*W(6)
      HTLOUT=HTLOUT1        *W(7)+HTLOUT2*W(8)
      IF (NTYPE.EQ.0) GO TO 58
      CALL ENTV(1,INSTGO,HTVIN1,  DH)
      CALL ENTV(1,INFDO,HTVIN2,DH)
      CALL ENTV(1,    OUTSTO,HTVOUT1,DH)
      CALL ENTV(1,OUTPRO,HTVOUT2,DH)
      HTVIN=HTVIN1          *W(1)+HTVIN2*W(2)
      HTVOUT=HTVOUT1        *W(3)+HTVOUT2*W(4)
   58 CONTINUE
      CALL CPLI(1,INSTGA,CP5)
      CALL CPLI(1,INFDA,CP6)
      CALL CPLI(1,OUTSTA,CP8)
      CALL CPLI(1,OUTPRA,CP7)
      IF (NTYPE.EQ.0) GO TO 59
      CALL CPVA(1,INSTGO,CP1)
      CALL CPVA(1,INFDO,CP2)
      CALL CPVA(1,OUTPRO,CP3)
      CALL CPVA(1,OUTSTO,CP4)
   59 CONTINUE
      CALL DENL(1,INSTGA,RHO5)
      CALL DENL(1,INFDA,RHO6)
      CALL DENL(1,OUTSTA,RHO8)
      CALL DENL(1,OUTPRA,RHO7)
      IF (NTYPE.EQ.0) GO TO 56
      CALL DENV(1,INSTGO,RHO1)
      CALL DENV(1,INFDO,RHO2)
      CALL DENV(1,OUTPRO,RHO3)
      CALL DENV(1,OUTSTO,RHO4)
   56 CONTINUE
      IF (NTYPE.NE.0) GO TO 69
      CALL  DENL(1,INSTGO,RHO1)
      CALL  DENL(1,INFDO,RHO2)
      CALL  DENL(1,OUTPRO,RHO3)
      CALL  DENL(1,OUTSTO,RHO4)
      CALL  ENTL(1,INSTGO,HTVIN1,DH)
      CALL  ENTL(1,INFDO,HTVIN2,DH)
      CALL  ENTL(1,OUTSTO,HTVOUT1,DH)
      CALL  ENTL(1,OUTPRO,HTVOUT2,DH)
      CALL  CPLI(1,INSTGO,CP1)
      CALL  CPLI(1,INFDO,CP2)
      CALL  CPLI(1,OUTSTO,CP4)
      CALL  CPLI(1,OUTPRO,CP3)
   69 CONTINUE
C
      CALL LAMB (1,OUTPRA,HTVAP)
      LIN=W(5)+W(6)
      W(7)=S(1,JCONT,3)
      LOUT=W(7)+W(8)
      CPLIN=(W(5)*RHO5*CP5+W(6)*RHO6*CP6)/LIN
      CPLOUT=(W(7)*RHO7*CP7+W(8)*RHO8*CP8)/LOUT
      IF (VOL(1).EQ.0.0) GO TO 41
      VIN=W(1)+W(2)
      VOUT=LIN+VIN-LOUT
      W(4)=VOUT-W(3)
      CPVIN=(W(1)*RHO1*CP1+W(2)*RHO2*CP2)/VIN
      CPVOUT=(W(3)*RHO3*CP3+W(4)*RHO4*CP4)/VOUT
   41 CONTINUE
      TOUT=HTLOUT/(CPLOUT)
      HTEXT=HCOEF*HAREA*(TEXT-TLOUT)
      DO 40 I=1,NCOMP
      XIN(I)=(X(I,5)*W(5)+X(I,6)*W(6))/LIN
      XOUT(I)=(X(I,7)*W(7)+X(I,8)*W(8))/LOUT
```

```
      IF(VOL(1).EQ.0.0) GO TO 43
      YIN(I)=(X(I,1)*W(1)+X(I,2)*W(2))/VIN
      YOUT(I)=(X(I,3)*W(3)+X(I,4)*W(4))/VOUT
   43 CONTINUE
   40 CONTINUE
C
C          COMPUTATION OF DERIVATIVES
C
      YOUT(NEQ)=TOUT
      YOUT(NEQ+NEQ)=VOL(2)
      IF(VOL(1).EQ.0.0) GO TO 61
      IF(NTYPE.EQ.0) GO TO 45
C
C          EXTRACTION SECTION
C
C          CONVERT TO SOLVENT FREE VARIABLES
C     ****  U-PU EXTRACTION
      CALL CONVT1(X,T,XTBP,DENOMA,DENOMO)
C     ****
C          EQUIL COMPOSITION OF ORG PHASE
C
      DO 42 I=1,NCOMP
      XSLF(I)=X(I,8)
   42 CONTINUE
      CALL ORGPH(TOUT,XTBP,XSLF,DISCO)
      DO 50 I=1,NCOMP
   50 YIDEAL(I)=XOUT(I)*DISCO(I)*DENOMO/DENOMA
      GO TO 46
C
C     VAPOR-LID SECTION
C
   45 CONTINUE
      CALL BOIL(TLOUT,XOUT,YIDEAL)
   46 CONTINUE
      DO 51 I=1,NCOMP
      TRANS(I)=YIDEAL(I)-YOUT(I)
      DERY(I)=(VIN*YIN(I)-VOUT*YOUT(I)+KEXT* TRANS(I))/(VOL(1)-VOL(2))
      DERX(I)=(LIN*XIN(I)-LOUT*XOUT(I) -KEXT*TRANS(I))/VOL(2)
   51 CONTINUE
      DERY(NEQ)=(HTLIN+HTVIN-HTLOUT-HTVOUT+HTEXT-HTVAP)/(CPLOUT*RHO7*
     1VOL(2)+CPVOUT*RHO3*(VOL(1)-VOL(2)))
      SIGTRAN=0.0
      DO 53 I=1,NCOMP
      SIGTRAN=SIGTRAN+TRANS(I)/RHO7
   53 CONTINUE
   61 CONTINUE
      IF (VOL(2).NE.0.0) GO TO 63
C
C     STIRRED TANK SECTION
C
      DO 62 I=1,NCOMP
      DERY(I)=0.0
      DERY(I)=(LIN*XIN(I)-LOUT*XOUT(I))/VOL(2)
   62 CONTINUE
      DERY(NEQ)=(HTLIN -HTLOUT+HTEXT)/(CPLOUT*VOL(2))
   63 CONTINUE
      TOUT=YOUT(NEQ)
C
C        CALCULATE RETURN VALUES
C
      DO 70 I=1,NCOMP
      XD(I,7)=DERX(I)
      XD(I,8)=XD(I,7)
      XD(I,3)=DERY(I)
      XD(I,4)=XD(I,3)
   70 CONTINUE
      TD(7)=DERY(NEQ)
      TD(8)=TD(7)
      TD(3)=TD(7)
      TD(4)=TD(7)
C
C     RENAME STREAM VARIABLES AND CONVERT TO MASS CONCENTRATIONS OF U-PU
C
      DO 91 J=3,4
```

```
4454        XXX(7,J)=XD(4,J)
4455        XXX(6,J)=XD(3,J)
4456        XXX(5,J)=XD(2,J)*239.0
4457        XXX(4,J)=XD(1,J)*238.0
4458        XXX(3,J)=0.0
4459        XXX(2,J)=TD(J)
4460        XXX(1,J)=0.0
4461     91 CONTINUE
4462        DO 92 J=7,8
4463        XXX(7,J)=XD(4,J)
4464        XXX(6,J)=XD(3,J)
4465        XXX(5,J)=XD(2,J)*239.0
4466        XXX(4,J)=XD(1,J)*238.0
4467        XXX(3,J)=0.0
4468        XXX(2,J)=TD(J)
4469        XXX(1,J)=0.0
4470     92 CONTINUE
4471 C
4472 C      CALCULATE OUTPUT TO STREAM VARIABLE FILE
4473 C
4474        IF (OUTPRO.EQ.0) GO TO 100
4475        S(1,OUTPRO,3)=W(3)
4476    100 IF (OUTSTO.EQ.0) GO TO 101
4477        S(1,OUTSTO,3)=W(4)
4478    101 IF (OUTPRA.EQ.0) GO TO 102
4479        S(1,OUTPRA,3)=W(7)
4480    102 IF (OUTSTA.EQ.0) GO TO 103
4481        S(1,OUTSTA,3)=W(8)
4482    103 CONTINUE
4483        DO 88 I=1,NC3
4484        IF(OUTPRO.EQ.0) GO TO 83
4485        S(2,OUTPRO,I+2)=XXX(I,3)
4486     83 IF(OUTSTO.EQ.0) GO TO 86
4487        S(2,OUTSTO,I+2)=XXX(I,4)
4488     86 IF(OUTPRA.EQ.0) GO TO 87
4489        S(2,OUTPRA,I+2)=XXX(I,7)
4490     87 IF(OUTSTA.EQ.0) GO TO 88
4491        S(2,OUTSTA,I+2)=XXX(I,8)
4492     88 CONTINUE
4493        EP(IM,1)=S(1,OUTSTO,7)*VOL(1)+S(1,OUTSTA,7)*VOL(2)
4494        RETURN
4495        END
4496        SUBROUTINE BOIL(TOUT,XOUT,XIDEAL)
4497        COMMON /PROP/MW(6),CPL(6,6),CVA(6,6),ENT(6,6),ENV(6,6),LAM(6,6)
4498       1,VAP(6,3),WI(6,6),DNL(6,6)
4499        COMMON /PTAB/TREF,R
4500        DIMENSION XOUT(10),YIDEAL(10),PV(   10)
4501        PVT=0.0
4502        XT=0.0
4503        T=TOUT+TREF
4504        DO 1 I=1,NCOMP
4505        PV(I)=EXP(VAP(I,1)+VAP(I,2)/(VAP(I,3)+T))
4506        PVT=PVT +PV(I)*XOUT(I)
4507        XT=XOUT(I)+XT
4508     1 CONTINUE
4509        DO 2 I=1,NCOMP
4510        YIDEAL(I)=PV(I)/(PVT*XT)
4511     2 CONTINUE
4512        RETURN
4513        END
```

REFERENCES

1. J. R. Barney, R. S. Ahluwalia, and A. I. Johnson, *DYNSYS 2.0 Users Manual*, Univ. of Western Ontario, London, Ontario (1975).
2. *General Purpose Simulation System/360 OS and Do 5 Revision 2 Users Manual*, IBM Corporation, White Plains, N.Y., No. SH20-06940 (1969).
3. A. A. B. Pritsker, *The GASP IV Simulation Language* (John Wiley and Sons, New York, 1974).
4. P. J. Kiviat, R. Villanueva, and H. Markovitz, *The Simscript II Programming Language* (Prentice-Hall, Englewood Cliffs, N.J., 1969).
5. R. L. Motard, *PRODYC—A Simulation Program for Process Dynamics and Control—Users Guide*, Dept. of Chem. and Petr. Eng., Univ. of Houston, Texas (1970).
6. L. A. B-Lopez, *DYSCO: An Iterative Executive Program for Dynamic Simulation and Control of Chemical Processes*, Ph.D. thesis, University of Michigan, Ann Arbor, Michigan (1974).
7. P. G. Ham, W. D. Seider, and R. L. Klaus, "The Transient Analysis of Integrated Chemical Processes", in *Proc. of 72nd Nat. A.I.Ch.E. Meeting, St. Louis, 1972.*
8. R. G. E. Franks, *Modelling and Simulation in Chemical Engineering* (John Wiley and Sons, New York, 1972).
9. *Continuous System Modelling Program Users Manual*, IBM Corporation, White Plains, N.Y., Application Program GH20-0367-4 (1971).
10. *Mimic—A Digital Simulation Language Reference Manual*, Control Data Corp., Minneapolis, Minnesota, Publ. No. 44610400, Rev. D. (1970).
11. S. M. Morris, *LEANS: Lehigh Analog Simulator* (Lehigh University, Bethlehem, Pennsylvania, (1973).
12. C. W. Gear, *Numerical Initial Value Problems in Ordinary Differential Equations* (Prentice-Hall, Englewood Cliffs, N.J., 1971).
13. C. W. Gear, "DIFSUB for Solution of Ordinary Differential Equations," *Comm. of the ACM 14*, 185 (1971).
14. A. C. Hindmarsh, *GEAR: Ordinary Differential Equation System Solver*, Lawrence Livermore Laboratory, Rept. UCID-30001, Rev. 3 (1974).
15. C. L. Miller, J. E. Hammelman, G. M. Borgonoui, *Dynamic Process Model of a Plutonium Oxalate Precipitator*, Lawrence Livermore Laboratory, Rept. UCRL-13812 (1977).

MLD