Computer-Aided Design and Simulation of Chemical Plants.

PEETPACK, A Non-Proprietary Flowsheeting Program.

By

Norman Peters

A Thesis Presented For The Award Of The Degree Of DOCTOR OF PHILOSOPHY, At The University Of Aston In Birmingham.

Department of Chemical Engineering. June 1974.

Supporting Publications :

1- "An appraisal of the use of PACER, GEMCS and CONCEPT for chemical plant simulation and design."
The Chemical Engineer, No. 284, March 1974, P 149.

2- "PEETPACK, a new non-proprietary flowsheeting program"
Sent for refereeing and publication in The Chemical Engineer.

3- "Data reconciliation by flowsheeting program"
Submitted for presentation and publication following the International Conference and Exhibition-Computers in Engineering and Building Design, Imperial College, September 1974.

4- "PEETPACK, a non-proprietary simulation program"
To be submitted for publication in Process Engineering in July/August 1974.

Public Presentations in Symposia :

1- "A critical appraisal of PACER, GEMCS and CONCEPT"
European Federation of Chemical Engineers Symposium, Paris, 24 April 1973.

2- "PEETPACK, a new non-proprietary flowsheeting program" First Annual Research Symposium, sponsored by the Science Research Council, Imperial College, London, 10 April 1974.

3- "A discussion on PACER, GEMCS and CONCEPT" Quarterly meeting of the C.I.Ch.E.E. committee, Cambridge University, 25 March 1973.

4- "An appraisal of PACER, GEMCS and CONCEPT" U.M.I.S.T. Departmental Symposium, 29 May 1973.

An Appraisal of the use of PACER, GEMCS, and CONCEPT for Chemical Plant Simulation and Design

By N. Peters† and P. E. Barker (FELLOW)†

It is becoming increasingly difficult to appraise the growing number of chemical plant simulation programs, and to assess their advantages and disadvantages for different types of users with different aims and requirements. In this paper, three well-known or easily available programs are compared in detail from the point of view of data inputting, recycle loop solution, computation phase and print-out. The three programs chosen are PACER, GEMCS, and CONCEPT. The computer simulation of an ethane-ethylene distillation plant using these three programs illustrates the discussion. The paper also presents a critical appraisal of the programs based on the requirements of the undergraduate student, the research postgraduate and the practising engineer.

Introduction

In a recent survey[1] of flowsheeting programs for the steady-state simulation and design of chemical plants, some 30 industrial and academic programs were mentioned and briefly described. The rapid growth in number and types of these programs, since the first one was mentioned in the literature in 1963,[2] and the proprietary nature of most of them have made their critical appraisal and assessment as to their advantages and disadvantages for various groups of users with different requirements quite difficult.

Following an introductory discussion on flowsheeting and flowsheets in general, three well-known or easily available programs are described, and a critical appraisal is presented from the point of view of three different types of users: the undergraduate student, the postgraduate researcher and the practising engineer. These programs are:

PACER—One of the first and best known academic programs.

GEMCS—A non-proprietary and one of the newest and most compact programs.

CONCEPT—The only interactive complete package available to interested users in Britain.

The solution of a typical example using all three flowsheets illustrates the description.

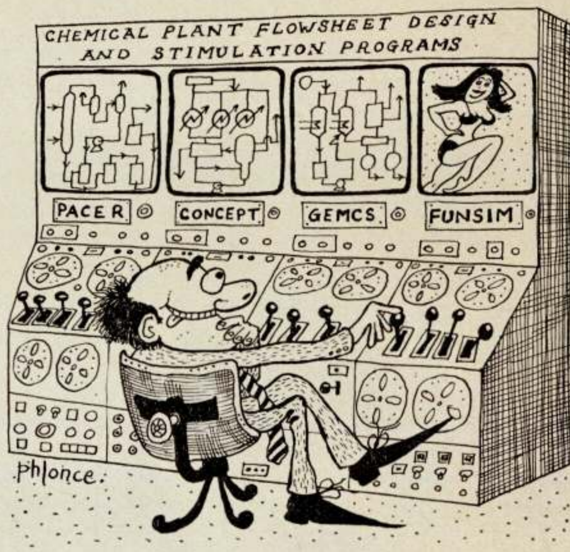
A General Description of Flowsheeting and Flowsheets

The basic function of flowsheeting programs is the calculation of steady-state heat, mass and energy balances in chemical plants, although there exists a different type of flowsheet that simulates start-ups, shut-downs, and transient states.[3] A number of publications[4–7] have described at various times the aims and features of flowsheeting, but the latest series of articles by Kehat and Sacham[8] gives the most recent picture and provides also a list of 115 references treating the various features of flowsheeting.

The main advantage of flowsheets is that the iterative calculation of large plants containing many recycle loops is automatically performed for the user. Large flowsheeting packages which include unit models and thermo-physical property generators enable the engineer to simulate a complete plant with relatively little effort and practically no knowledge of programming. But with more compact and basic programs the user must provide most of the models and physical data. The structure of flowsheets may be divided into six sections but only the largest and most complete (usually industrial) programs will incorporate them all:

(1). The Data Input (interactive or batch-mode). The main data required are the process flow diagram (called process matrix because of its usual matrix-type storage), the list of parameters for each unit in the plant (the equipment parameter matrix), the physical description of the plant streams (flow, temperature, pressure, composition, *etc.*; the stream matrix), the components present

†*Chemical Engineering Department, University of Aston in Birmingham, England.*



in the plant (optional) and the convergence control parameters. Additional information pertinent to each program helps consolidate the above data.

(2). The Calculation Order Finder (optional). The process diagram is partitioned into a number of recycle sets (single or nested loops), and the minimum number of streams that tear the system open, subject to certain user-imposed constraints, are used as recycle streams (iterate streams, which are continuously checked for convergence). These streams and the plant feeds are then followed through the plant to obtain the calculation order.

(3). The Calculation Executive. This section calls the various models to simulate the different plant units. It provides them with the necessary information to perform the heat and mass balances and re-stores the updated streams and parameter values. It may also accelerate the convergence of the solution by applying convergence promotors to the iterate streams. At convergence it calls the data print-out section.

(4). The Data Print-out. Intermediate print-out during calculation may be requested. Depending on the program it will generally consist either in the stream and parameter values entering each model calculated, or in a complete stream and parameter matrices print-out at suitable calculation intervals. If instructed, the model may also generate some informative print-out. The final print-out varies with the type of program used. In batch-mode operation the final complete results will normally resemble the intermediate print-out: either a full matrices print-out or a last complete plant computation with input and output stream and parameter print-out. In addition any energy and/or design and costs estimates will be produced. In

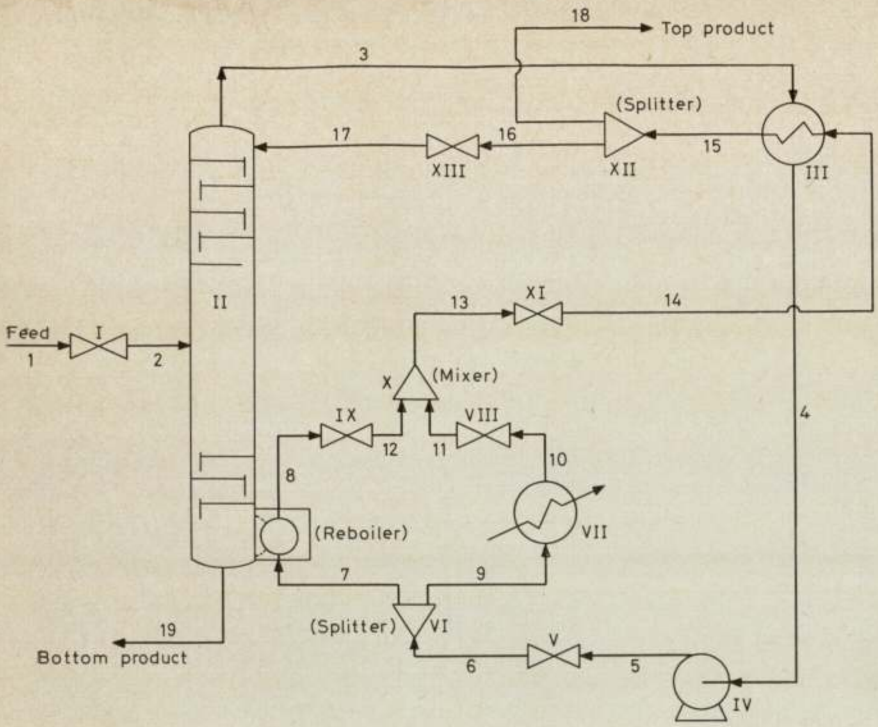


Fig. 1.—*Flow diagram of the splitter plant*

interactive mode, the same print-out as in batch-mode may be sent to the line printer, or the print-out of individual streams and unit parameters may be asked for at the teletype.

(5). The Unit Models. A set of mathematical relationships describing the heat and mass balances operate on the input stream information to produce the output stream values subject to the unit's parameters and the component properties. Each of the four sets of data mentioned here is available explicitly for the unit under consideration in clearly labelled and separate matrices as supplied by the executive. Unit models are not supplied with all flowsheets or for unusual units. The user will normally have to provide his own models.

(6). The Thermo-Physical Data Bank. All flowsheets need, but few provide, a library of routines that calculate thermodynamic data such as vapour–liquid equilibrium constants, dew and bubble points, enthalpies, *etc.* or physical properties such as pseudo-critical data, densities, viscosities, *etc.* Some flowsheets allow users to input their own thermophysical data which will normally be required for exotic or complex components.

The omission of some of these sections from more compact programs is more readily understood when flowsheeting programs are later analysed in the context of the type of user they are intended for.

The Programs to be Compared

The three programs chosen for comparison encompass a variety of types of users and have specific advantages for each type.

PACER, the best known flowsheeting program was developed initially by H. A. Mosler at Purdue University in 1964. It was then improved by P. T. Shannon at Dartmouth College, New Hampshire, and by a team of professors and P. T. Shannon at McMaster University (Canada), from whom one of the authors (P. E. Barker) of this article acquired the 1966 McMaster version. It should be mentioned that a new commercial version, PACER 245, now exists in the USA.[9] The 1966 version available to us is a batch process executive program with few unit models and no physical data package. It is made up of four sections: the data input section, a calculation order finder, the calculation phase, and an intermediate and final print-out section.

GEMCS is one of the most compact executives, and is presently enjoying a wide academic use. It was developed by Professor A. I. Johnson and associates in 1968 at McMaster University, Canada, from whom it is now available.[10] It can be used either batchwise or interactively.[11] Like PACER, from which it was derived, but to which it bears little resemblance, it is an executive program but does not incorporate a calculation order finder. It still has the three other sections but these are very different from PACER's.

CONCEPT was initially developed in the late 1960's by a number of postgraduate students under the supervision of Mr H. P. Hutchison at Cambridge University (UK), and it is now being improved and offered for use to paying users by the Computer Aided Design Centre (CAD) in Cambridge. This complete package comprises general but accurate unit models and a large thermodynamic data bank, and it may be accessed (interactively only) either by teletype terminal or Cathode Ray Tube and light pen.[12]

An Example Solved on All Three Programs

The simulation of an ethane-ethylene distillation plant, shown diagramatically in Fig. 1, on all three programs helps illustrate their differences in data input and calculation order finding. In this example, a liquid mixture of ethylene (44 mol %), ethane (54 %), and propane (2 %) is fed to a distillation column of 55 actual trays, and the top vapour product is superheated in a heat exchanger by the effluent of the heating-side of the reboiler, then compressed and heated further and partly used to reboil the column bottoms. The other part of the compressed vapour is sub-cooled and added to the reboiler effluent to control the stability of the plant. After heating the top vapour this stream is divided into recycle back to the column and product. The column bottom product is not treated.

To facilitate the simulation, a few modifications and simplifications were introduced:

(1). The propane in the feed was ignored and the feed (45 % ethylene, 55 % ethane) was taken as saturated liquid at the plant inlet pressure (345 lbf/in^2 abs.), then throttled down to the column pressure (215 lbf/in^2 abs.).

(2). The number of trays in the column was reduced to give results comparable with the plant data and the trays were considered ideal (40 trays).

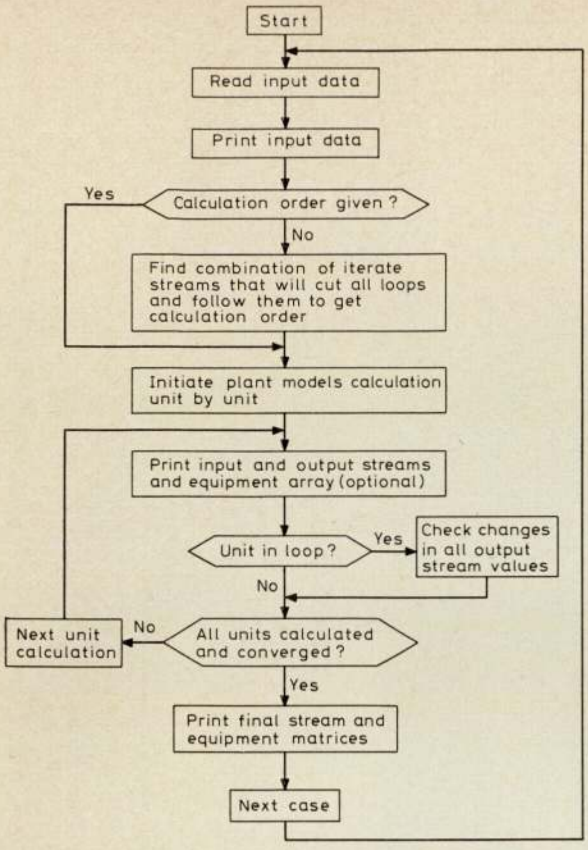


Fig. 2.—*The simplified PACER flow chart*

(3). The pressure drop in the pipes, control valves, and heat exchangers were lumped together and represented by a few throttle valves.

Some of the relevant plant data are given in Table III.

The Comparison of the Programs

PACER (1966 *McMaster version*)

The four sections of its executive occupied about 18 000 words on an ICL 1904. When the unit models were added, its total size exceeded 32 000 words, but even with better programming the size could not be reduced below 28 000 words. This is mainly due to the large number of large matrices for data storage.

THE DATA INPUT

Three main sets of data are needed:

The process matrix bears in each row the unit number, its model and plant identifiers and its input (positive) and output (negative) stream numbers.

The equipment parameter matrix carries in each row the unit number and the parameters required to fully describe this unit.

The stream matrix contains the physical description of the streams (stream number, flag, flow, temperature, pressure, enthalpy, composition). The flag specifies the nature of the streams (feeds, recycles, unknown). All unknown or unspecified streams must be read in with zeroes for their description.

TABLE I.—*Part of PACER's Process Matrix*

Unit Number	Model Name	Plant Identifier	Associated Streams					
1	THROTL		1	−2	0	0	0	0
2	DISCOL	—	2	17	7	−3	−8	−19
3	HEXCH	C–113	3	14	−4	−15	0	0
4	COMPRS	J–44	4	−5	0	0	0	0
5	THROTL	—	5	−6	0	0	0	0
6	SPLITR	—	6	−7	−9	0	0	0

All arrays in all matrices are of preset maximum length which is set by reading in "maximum size" variables. This, however, does not supersede the internal fixed *dimension* of the fixed size matrices.

Two other matrices for extra stream and unit descriptions are provided but rarely used. All streams and units must be given different numbers as identifiers. Alphanumeric names are not accepted.

The calculation order input (optional) may be read in array form showing each unit number in the order followed by a status flag (in a sequential set, recycle set or end of loop), thus by-passing the order finder routine. Other data controlling the iterations and suggesting preferred streams as starting streams in the order finding are also required.

Once all the data are read in, a well documented echo is printed out. Tables I and II illustrate part of the input data for the solved example.

THE CALCULATION ORDER

When not read in, a stream connection matrix showing the source and sink of each stream is prepared and is used to find the initial sequential sets of units which are first calculated. The first unit with an unknown input (recycle) ends this sequence. To solve the remaining units, whether in loops or not, the program chooses all possible sets of one, then two, then three unknown (iterate) streams, assumes them to be known and tries to follow them through the plant. A list is made up of all units met with no unknown inputs. For each new addition to the list, all uncalculated units have their inputs checked and their output streams are flagged "known". The calculation order is the path found by the first set of streams to place all unknown units in the list. The method will not solve plant involving more than three recycle streams. Preferred streams may be proposed to start the search in order to accelerate it, but all sets of one, then two streams must still be tried before sets of two, then three streams are used.

Two iterate streams had to be "cut" to solve the recycle loops in this plant: streams 3 and 13. Preferred streams (4 and 14) were proposed to start the search and these were accepted, but the order was slightly modified.

THE CALCULATION PHASE

Every type of unit in a plant is given a CALL statement, the set of which is in an arbitrary sequence in the model calling routine. A set of input data must specify the model used for each unit in an array showing the model name in the location corresponding to the unit number. Units in sequential sets are calculated before the ordering of recycle sets is found; then when the order is obtained, each unit in it is called in its order and calculated. In a loop the relative change in all properties of all its streams is checked against a preset tolerance for convergence, but no convergence accelerator is

TABLE II.—*Feed Stream and Initial Guesses*

Stream Number	Flag	Flow (lb mol/h)	Temperature (°R)	Pressure (lbf/in^2 abs.)	Enthalpy (Btu/lb mol)	Ethylene Fraction (mole fractions)	Ethane Fraction (mole fractions)
1	1	511	440	334	2400	0.45	0.55
4	0	1325	452	215	6400	0.95	0.05
17	0	1085	430	215	2400	0.95	0.05

1.0 (unit no.)	1.0 (type)	16.0 (array length)	0.0	0.0
1.0 (no. of inputs)	1.0 (stream no.)	0.0	0.0	0.0
1.0 (no. of outputs)	2.0 (stream no.)	0.0	0.0	0.0
215.0 (outlet pressure)				
2.0 (unit no.)	2.0 (type)	26.0 (array length)	0.0	0.0
3.0 (no. of inputs)	2.0	7.0 (stream numbers)	17.0	0.0
3.0 (no. of outputs)	3.0	8.0 (stream numbers)	19.0	0.0
40.0 (no. of plates)	20.0 (feed plate)	215.0 (colm. pressure)	235.0 (tops flow rate)	235.0 (vapour flow rate) (on plate 1)
235.0 (vapour flow rate on last plate)	0.95 (tops composition)	0.05	1.0 (key components)	2.0
4.60 (reflux ratio)				

Fig. 3.—*Some of GEMCS's equipment arrays*

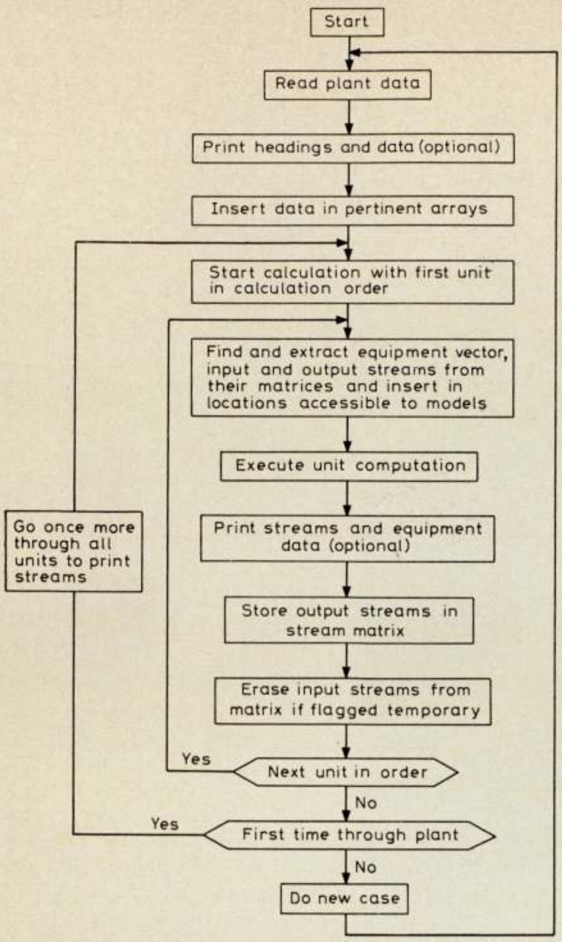


Fig. 4.—*The GEMCS flow diagram*

provided. For this plant seven models were needed and the convergence tolerance of 0.1% was achieved in five passes only, around the plant, because of the good initial guesses shown in table II.

THE PRINT-OUT SECTION

The optional intermediate print-out may consist either of the streams in and out of each unit and its parameters or of the complete stream and parameter matrices for every specified number of loops. Print-out from the user-supplied models depends on the programmer, but cannot be suppressed, except by re-programming. The amount of print-out in the first case is very large, especially for long problems. A print-out of the main steps followed in the calculation order finder may also be given. The final print-out consists of the final stream and parameter matrices only.

A simplified flow chart of this long and complicated executive is shown in Fig. 2.

GEMCS

The executive program can occupy less than 5000 words of core and plants of fair complexity (10 to 15 units) have been run in overlay mode on an IBM 1130 (8K) computer with disc backing. Its three main sections are:

THE DATA INPUT

Four sets of data are required:

The first array provides the sequence in which units are to be calculated. Recycle loops are not shown explicitly, but each set of nested loops is ended by a convergence controller unit which must be included explicitly as any other plant unit (its use is explained below).

The second set arrays are the known and guessed streams. Unknown streams need not be input. These are arranged in a variable size matrix. Plant streams can be kept in this matrix either temporarily until they are used, or permanently. Their status is defined by the third set of data, the stream flags.

The equipment parameter arrays are read in and stored "head-to-tail" in one cumulative array. Their input and retrieval from this array is controlled by a special routine. The process matrix is part of the equipment parameters. The number of input and output streams and these stream numbers are stored in fixed locations in the parameter arrays, Fig. 3.

Each set of arrays is preceded by the number of arrays it contains. Printing flags fed in at the top of the data monitor the amount and type of data echo and intermediate print-out.

THE CALCULATION PHASE

A pointer pointing at the unit numbers in the calculation order decides on the unit to be calculated. This pointer starts at the first unit in the order and advances by one position after each calculation. In recycle loops, when the convergence controller is reached, the relative change in all properties of one preset stream are checked against a preset tolerance, and until convergence is reached the pointer is returned backwards by the number of units in the loop, and the calculations repeated. This controller is used like a unit model and must be given a unit number, its relevant parameters and a CALL statement. Units are called according to their type (a parameter) rather than their plant order. In the example eight types of units are present: VALVE, DISCOL (distillation column), HEXCH (heat exchanger), COMPRS (compressor), SPLITTER, MIXER, and CONTLI (convergence controller, given a unit number of 14). The order of units is 1, 4, 5, 6, 7, 8, 2, 9, 10, 11, 3, 12, 13, 14. The iterate stream checked is 4 and the calculation is repeated for the sequence 4 to 14.

THE PRINT-OUT PHASE

The data echo is optional and depends on the setting of certain print switches. Other print switches allow or suppress the printing of the stream matrix and/or the input streams to a unit and its parameter array at each unit calculation. To obtain the final print-out, the program recalculates all the units again, each once, and prints out their output streams. Print-out from user-provided models may be monitored by the print switches.

This program is simple and easy to understand as it was written for teaching and as a basis for individual elaborate flowsheets.

An associate program (ORDER) helps find up to three recycle streams, based on the PACER method, but does not provide the calculation order.

Figure 4 shows the GEMCS flow diagram.

CONCEPT

CONCEPT has extra design and costing facilities in addition to simulation capability. It was designed to operate in a number of interactive segments occupying 12 000 to 20 000 words on an ATLAS computer. The extra facilities are entered only if expressly requested by the user. Extensive model and data generation routines complement the sections described below.

THE DATA INPUT

The plant flowsheet description may be entered interactively by simple responses to the program. All data are checked on input against existing information. On completion of this input the programs LOOPFINDER and ITFINDER are used to determine the calculation order. One of two courses may then be taken. If most of the data has been stored in files, a procedure TALKBACK requests these files and checks them for omissions. Subsequently either a calculation is initiated or a list of errors is typed out, and command is transferred to the DIALOGUE. The second course is to enter the interactive DIALOGUE immediately. There, a number of recognised commands coupled to a file, unit, stream, or component name enable the user to input, change, or examine any data or results. These commands may be used repetitively before and after the calculation phase. Some commands are illustrated in Fig. 5.

```
INTERACTIVE FLOWSHEET SYSTEM
ENTERED - WAIT FOR '——→'
——→ TVI
NO. OF I/PS = 1
NO. OF O/PS = 1
I/P NO. 1 :  1
O/P NO. 1 :  2
——→ DC2
NO. OF I/PS = 3
NO. OF O/PS = 3
I/P NO. 1 :  2
I/P NO. 2 : 17
I/P NO. 3 :  7
O/P NO. 1 :  3
O/P NO. 2 : 19
O/P NO. 3 :  8
```

Fig. 5 (a).—*Part of CONCEPT's Interactive Input*

```
? INPUT STREAM 1
STREAM 1
FLOWRATE              = 511
TEMPERATURE DEGF    - -20
PRESSURE     PSIA   = 345
VAPOUR FRACTION     = 0
ENTHALPY BTU/LBM    = 0
WATER               = 0
ETHYLENE            = .45
ETHANE              = .55
? INPUT ALL GUESSES
GUESSED STREAMS
- - - - - - -
LDB NO. 1
STREAM 17
FLOWRATE            = 1085
TEMPERATURE DEGF    = -40
PRESSURE     PSIA   = 215
VAPOUR FRACTION     = 1
ENTHALPY BTU/LBM    = 0
WATER               = 0
ETHYLENE            = .95
ETHANE              = .05
STREAM 4
FLOWRATE            = 1325
TEMPERATURE DEGF    = -8
PRESSURE     PSIA   = 215
VAPOUR FRACTION     = 1
ENTHALPY BTU/LBM    = 0
WATER               = 0
ETHYLENE            = .95
ETHANE              = .05
```

Fig. 5 (b).—*Example of Stream Data Input*

```
? INPUT ALL PARAMETERS
PARAMETERS FOR DC2
HOW MANY ? 8
1 . 40
2 . 20
3 . 235
  . 235
5 . 4.6
6 . .95
7 . .05
8 . 1
PARAMETERS FOR HE3
HOW MANY ? 7
1 . 1
2 . 0
3 . 0
4 . -8
5 . 0
6 . 0
7 . 0
```

Fig. 5 (c).—*Example of Parameter Input*

```
? INPUT ALL S/RS
TYPE THE S/R NAME AFTER
THE UNIT NAME
IN1 - SOURCE
TV1 - TVALVE
DC2 - DISCOL
HE3 - HEATEX 1
K4  - COMPRESS
TV5 - TVALVE
```

Fig. 5 (d).—*Example of Model Routines Allocation*

THE CALCULATION ORDER FINDER

The order is found in three steps.[13] The first step (the LOOPFINDER) locates all closed loops in the flow chart by tracing backwards the input to each unit through all the units it passes through until either a plant feed or the initial unit is met. In the second case a recycle loop is found. All plant streams are traced backwards to locate all possible loops. A unit may be part of more than one recycle loop if by tracing back its different feeds they lead back to it after passing by different units. The second step (the ITFINDER) locates the iterate streams by eliminating in each loop the streams which are common to the least number of other loops, or all except the first stream in loops which share no common streams with other loops. The number of streams is thus reduced to the minimum number of streams cutting all loops. These become the iterate streams unless the user prefers to change them, in which case the new streams are checked for sufficiency and other streams are found if the proposed streams do not cut all loops. In the third step, the iterate streams are followed one at a time through the plant as long as no unchecked stream (output from an unchecked unit) is met. The units met are checked and set in a list which forms the calculation order.

THE CALCULATION PHASE

The units are called by the executive according to their model's type rather than their plant numbers. At every pass through a complete set of nested loops, all properties of the iterate streams are checked for convergence, and convergence may be accelerated by a Wegstein-type[14] promoter. On achieving a successful run, a last pass through all units is affected to print stream information and design data control is then returned to the DIALOGUE where the use reexamines his result and perhaps changes the data and recalculates the plant. The calculation phase may be run off-line by "dumping" the data files and using OFFLINE commands. Although CONCEPT provides its own models and thermodynamic routines, users may easily interface to their own sets of routines for use independently or in conjunction with CONCEPT's libraries.

THE PRINT-OUT SECTION

Following a run, stream properties may be examined in the DIALOGUE or output along with the input data, parameters and computational sequence to the line printer. Intermediate print-out consists of the input streams to each unit calculated and the final print-out is the complete stream matrix. All output generated during a run is recorded in a log file which may then be listed on line printer or teletype. Print-out from users' routines may be printed at every unit calculation or at the final pass through the plant.

Detailed information about CONCEPT is confidential, but its simplified flow chart[15] is presented in Fig. 6.

THE SIMULATION RESULTS

PACER's and CONCEPT's iterate stream finders both suggested streams 3 and 13 as iterate streams, but the insensitivity of the column model to its reflux (stream 17) and the independence of the compressor model on its feed enthalpy, as opposed to the valve and the exchanger's need of more accurate feed guesses, suggested instead the choice of streams 4 and 17, which were accepted but which modified the unit calculation order from units (1, 11, 3, 4, 5, 6, 7, 8, 12, 13, 2, 9, 10) to units (1, 4, 5, 6, 7, 8, 2, 9, 10, 11, 3, 12, 13).

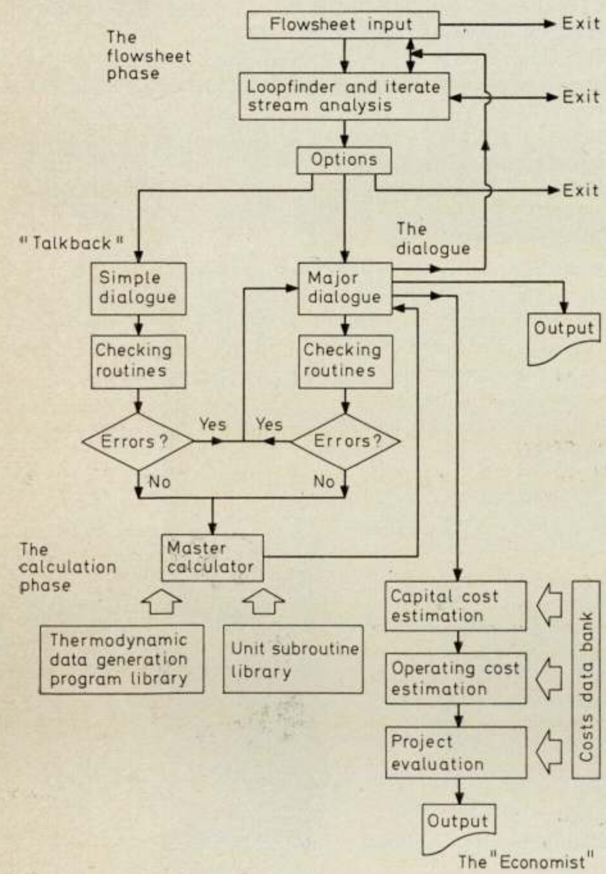


Fig. 6.—*The Concept Mark 3 System*

TABLE III.—*Plant Data and Simulation Results*

Property	Location	Data: From Survey	Data: From Simulation
Flows (lb mol/h)	Feed	511.0	511.0
	Tops	235.0	235.0
	Bottoms	276.0	276.0
	Throughout plant	1325.0	1316.0
	To Reboiler	1080.0	1200.0
Compositions (Ethylene mol %)	Feed	44.0	45.0
	Tops	95.0	94.0
	Bottoms	2.5	3.0
	Throughout plant	95.0	94.0
Temperatures (°R)	Feed	—	440.0
	Column tops	422.0	420.0
	Column bottoms	455.0	454.0
	Compressor inlet	452.0	454.0
	Compressor outlet	573.0	578.0
	Cooler outlet	439.0	439.0
	Reboiler outlet	465.0	463.0
	Liquid outlet from exchanger	444.0	443.0
Pressures (lbf/in^2 abs)	To plant	345.0	345.0
	In column (average)	215.0	215.0
	Compressor outlet	585.0	585.0
	Reboiler outlet	414.0	414.0
	Cooler outlet	457.0	457.0
	Reflux to column	215.0	215.0
Reflux ratio	To column	4.6	4.6
Heat loads (Btu/h)	In heat exchanger	—	0.48×10^6
	In cooler	—	5.52×10^6
	In reboiler	—	0.55×10^6
	In condenser	—	4.55×10^6

The enthalpy calculation packages used on PACER/GEMCS and on CONCEPT were different. PACER and GEMCS used the Redlich-Kwong equation of state iteratively for enthalpy calculations, and CONCEPT used this once through Yen and Alexander method.[16] All three programs used Chao-Seader's[17] vapour–liquid equilibrium method and the API[18] pure components properties. The three sets of simulation results were very similar so only one set is shown in Table III. The slight deviations from the test-run data are due to the neglect of propane in the feed and the deviations of the predicted properties from true values. The main discrepancy is in the flow-rate to the reboiler which had to be increased from 1080 lb mol/h to 1200 lb mol/h to meet the heat load in the reboiler. However, because of insufficient plant information, this discrepancy could not be explained.

The programs were run on different computers (ICL 1904 and ATLAS), and our models and enthalpy routines were much longer and less efficient than CONCEPT's, hence computation time could not be compared favourably. However, CONCEPT's convergence in four passes through the whole plant required only nine seconds of computing time on the ATLAS computer as opposed to over 100 seconds for GEMCS on ICL 1904.

A Critical Appraisal of the Program

User requirements in flowsheeting

The aims and objectives of the student, the postgraduate and the practising engineer when using flowsheet programs are very different, and each type of user expects to find in the program he uses the features that will help achieve his aims most effectively. Table IV summarises the main features which it is considered should exist in the program used by each of three categories of users. These features emphasise the learning process in modelling for students, the flexibility of the executive for the researchers and the ease of running problems for the engineer. The following appraisal is written in the light of these requirements, although it will be appreciated that these requirements represent the authors' personal viewpoint based on their computer simulation experience.

TABLE IV.—*Requirements in Simulation Programmes for Different Users*

The Undergraduate	The Postgraduate	The Engineer
Usually involved with:	Usually involved with:	Usually involved with:
The writing of models	The writing of design oriented models	Plant design
The study of parameters	The improvement of the executive	Plant simulation and optimisation
The study of initial guesses of operating conditions		
Requires:	Requires:	Requires:
An optional calculation order finder	Calculation order finder usually unnecessary	Most basic models to be available
Large intermediate print-out	Large intermediate print-out	A calculation order finder
Ease of data input	Short data input	Minimum print-out
Batch processing	Batch or interactive processing	Interactiveness
A library of routines	Internal re-ordering of calculation order should be possible	Ease of data input and update
	Referencing to other unit arrays	Fast access to results
	Small executive size	Immediate restart of run
		Limited interest in the executive

The Appraisal

PACER

PACER does provide all the requirements needed by the undergraduate and is the only program to offer an optional calculation order finder. However, its disadvantages far outweigh its advantages. The data input is long and requires a lot of unnecessary data specifications, such as all the unknown streams in the stream matrix, a complete specification of all matrix sizes, *etc.*, which could all be computerised. The calculation order finder, which relies on exhaustive searching, is very time-consuming and can only tackle up to three recycle streams. The large size of the executive (about 18K words) makes it difficult to use on small or medium (32K) computers. For the research postgraduate, it is so complicated, inflexible and very interrelated between its different sections, that it is uninteresting for development work. For the practising engineer, the lack of even the most basic models and physical data files and the expensive order finder make it very unattractive, since he would have to build up his complete libraries of data and models and ensure that all problems contain less than four recycle streams.

The version of PACER available to us has assisted in the field of flowsheeting development, but has since been superseded by far better programs for all classes of users.

GEMCS

GEMCS is ideally suited for both the student and the researcher. Its compact size (about 5K words), its short and modifiable input requirements and its simple structure make it easy to understand and use. Its lack of unit ordering facility seems preferred by academicians for teaching purposes, but the input order may be temporarily by-passed to repeat the calculations of short sequences of units simply by resetting the pointer to the unit calculated. This facility, along with that of referring to and changing other unit parameters, makes it very useful in optimisation and design work. A set of print switches helps monitor the amount and kind of intermediate print-out to the user's requirements.

GEMCS may seem unattractive to the engineer for the same reasons as PACER is, but it is a short and effective executive for tying together the models of a plant on which engineers will work for a long time. Once the models are written and the data prepared, the calculation order finder and the extra models and data in large programs, such as CONCEPT or PACER, become unnecessary thus making them, and hence the use of large computers, redundant.

CONCEPT

With the limited computer facilities in many Universities, and the difficulty in getting long intermediate print-outs for model checking on teletype terminals, CONCEPT's interactiveness is more of a hindrance than a help to undergraduates. Its overall design and the wide range of facilities it provides are intended much more for industrial applications rather than undergraduate teaching. Its high degree of sophistication leaves little scope for more postgraduate development and its large size makes it awkward as a research tool.

For industrial simulation CONCEPT is an excellent program. It provides sufficient accurate and general models and physical data for the simulation of plants of average complexity and its calculation order finder is very efficient. The data input section demands little preparation, and running plant simulations at different conditions requires the minimum possible data changes. Convergence promoters help to accelerate the calculations, and the final print-out is as concise as the user wishes to make it, thus allowing him to modify the data and re-run the simulation immediately. CONCEPT's new version offers a number of chemical unit operation design models and economic study and costing routines that can be accessed independently from the simulation executive following a plant simulation. Plant design optimisation is not yet available in the package.

Conclusion

The main sections and features of three well-established simulation programs (PACER, GEMCS and CONCEPT) have been briefly discussed and illustrated through the solution of a simulation example. Their critical appraisal showed that PACER is of interest only to undergraduate students, GEMCS is mainly of academic interest, but may have certain industrial applications, and CONCEPT is primarily of value as a good industrial simulation package.

Acknowledgments

The authors wish to express their gratitude to Dr M. E. Leesley and Dr P. Winter of the CAD Centre, Cambridge, and to Professor A. I. Johnson, Dean of Engineering Science, University of Western Ontario, for their help and co-operation in using the simulation programs and their general assistance with flowsheeting concepts.

References

[1] Flower, J. R., and Whitehead, B. D. *Chem. Engr, Lond.*, 1973, No. 273, p. 271.

Professor P. E. Barker was appointed a Chair in Chemical Engineering at the University of Aston in January, 1970. His research interests include production scale chromatography special separation techniques and computer aided process design. Mr N. Peters is a graduate of McGill University currently studying for a Ph.D. at Aston.

[2] Kessler, M. G., and Griffiths, P. R. *Proc. Am. Petrol. Inst.*, 1963, **43**, 49.
[3] Johnson, A. I. *Br. chem. Engng*, 1972, **17**, No. 3, p. 217.
[4] Evans, L. B., Stewart, D. G., and Sprague, C. P. *Chem. Engng, Prog.*, 1968, **64**, No. 4, p. 39.
[5] Johnson, A. I., *et al. Br. chem. Engng*, 1971, **16**, No. 10, p. 973.
[6] Johnson, A. I., *et al. Br. chem. Engng*, 1972, **17**, No. 1, p. 29.
[7] Flower, J. R., and Whitehead, B. D. *Chem. Engr, Lond.*, 1973, No. 272, p. 208.
[8] Kerat, E., and Sacham, M.
[9] Pacer 245. *Digital Systems Corporation.* (Hanover: New Hampshire).
[10] Johnson, A. I. *University of Western Ontario.* (London: Ontario).
[11] Johnson, A. I. *GEMCS Users Manual.*
[12] Forder, G. J., and Hutchinson, H. P. *Can. J. chem. Engng*, 1970, **48**, No. 1, p. 79.
[13] Forder, G. J., and Hutchison, H. P. *Chem. Engng Sci.*, 1968, **24**, 771.
[14] Wegstein, J. H. *Com. A.C.M.*, 1958, **1**, No. 6, p. 9.
[15] Computer Aided Design Centre. *CONCEPT: User Manual.* (Cambridge).
[16] Yen, L. C., and Alexander, R. E. *A.I.Ch.E.Jl*, 1961, **7**, No. 4, p. 334.
[17] Chao, K. C., and Seader, J. O. *A.I.Ch.E.Jl*, 1961, **7**, No. 4, p. 598.
[18] American Petroleum Institute. *Technical Data Book and Project* 44.

Received 17 *May*, 1973

PEETPACK, a new non-proprietary flowsheeting program

by

N. Peters and Professor P.E. Barker

Department of Chemical Engineering
The University of Aston, Birmingham 4, U.K.

Summary

There are over thirty flowsheeting programs presently in use or at various stages of development in North America and in Europe, however very few of them are non-proprietary and of these non contains a sufficiently wide or general library of unit models and physical property estimation routines or contains a calculation order finder which is particularly efficient. Many programs suffer also from a poor data interface with the user which makes their teaching or learning process difficult and time consuming.

PEETPACK is the contribution of the University of Aston towards fulfilling this need for a new flexible, complete and above all non-proprietary program. It contains general unit models and a large number of thermodynamic and physical property evaluation routines and its interactive or batch-mode operation is smoothly controlled by a number of Job control programs.

----oooOooo----

Introduction

The need and justification for a new flowsheeting program

In a recent survey (1) over 30 flowsheeting programs were mentioned and briefly discussed and to propose a new program would seem, at first thought, of little purpose. Unfortunately of these 30 or so programs only three or four are non-proprietary and easily obtainable in Britain (GEMCS (2), ESSPROSS (3) and UNICORN (4)) and of these none offers a sufficiently large library of general unit operation models or a library of thermodynamic or physical property evaluation routines. The calculation order finder, offered only in GEMCS merely helps the user locate up to three iterate streams and leaves him the task of following them through the plant to obtain the calculation sequence. The few complete flowsheeting packages made available to academic institutions for teaching purposes (CONCEPT(5) and FLOWPACK (6)) are offered only in pre-compiled object form with no listings (i.e. as black boxes) and at considerable cost.

The other justification in offering a new program is its teaching and technical contents. Through their use of a number of simulators (7), the authors have realised that some important and common deficiencies existed in many of them, mainly:

1. The poor data communication between the programs and the user.
2. The difficulty in tracing back errors in runs that fail.
3. The various technical limitations imposed by the lack of the most basic models and property data.
4. The difficulty in teaching the principles of flowsheeting and the use of the packages because of the limited program documentation.

In the light of these limitations the authors have developed a new program that would help solve many of these problems.

PEETPACK answers the requirements for a new flowsheeting program

PEETPACK (Process Engineering Evaluation Techniques Package) is Aston University's contribution towards fulfilling this need for a new flexible, complete and above all non-proprietary flowsheeting package. It was developed entirely on local resources to justify very specifically this last requirement.

The following criteria (8) have been closely considered and followed in its preparation:

1. The program must be made freely available to all interested users.
2. It must be constructed in a highly modular form and in a high level language to ease its installation, understanding and maintenance.
3. It must be adaptable to specific problems by ad hoc modular extension.
4. The input data structure must be flexible to allow a wide variety of small and large applications and to facilitate the information flow within the network of programs.

5. Algorithms and computational sequencing must be planned to minimise numerical sensitivity and convergence instability in iterative processes.

6. The models and thermo-physical property evaluation routines should be based on well-documented theoretical and numerical methods (especially for academic use).

7. Error trapping and tracing facilities should make the program run smoothly and help the user diagnose most error sources.

The shaping and writing of PEETPACK was also directly influenced by the critical appraisal of the use of PACER, GEMCS and CONCEPT (7), three of the best known or more easily available flowsheeting programs.

The general outline of PEETPACK

The aims set when writing PEETPACK, and listed above, were achieved through a study and redevelopment of the various constituents of conventional flowsheeting programs of the modular-type, but their overall structure was retained for three main reasons:-

1. It provides the most logical and obvious approach in dealing with modular simulation-orientated programs.

2. It is very easy to teach to students and to convey to practising engineers who are not familiar with flowsheeting techniques.

3. Flowsheeting programs remain on the whole simulation-orientated because of the lack of standardised plant design methods or approaches and because of the need for consistent heat and mass flow estimates before implementing design procedures.

The objectives were achieved in the following ways:

1. The ease of learning and use of the new package was achieved by writing completely new data input/output/update programs for both batch and interactive modes usage, and into which the users could input the minimum number of data in comprehensible alphanumeric form.

2. The high modularity of the whole package was achieved by splitting the various tasks in the package into a number of semi-independent or free-standing programs and dividing the task within each program into a number of fairly short routines each performing one duty only and relying on as few other routines as possible so that their replacement does not disturb the rest of the program.

3. The unit model library contains a large number of models of the basic physical process type such as heat transfer and separation processes, but a reactor model is also offered though chemical processes units are usually very process-specific and users are advised to use their own models.

4. The thermodynamic and physical property evaluation routines cover non-polar hydrocarbons and pure gases which obey closely the corresponding-state principle and are based on the best known methods, for the thermodynamic routines, and on the best methods available (9) for the physical property routines.

5. Error detection is facilitated through the incorporation of many warning and error messages in the models and property routines, and by displaying the state of all process streams at the time of failure in unsuccessful runs.
6. Modular extension to adapt new routines and models is possible through the creation of MACRO's that accept user-provided libraries of routines and which incorporate them into the package by reconsolidating its various constituents.

PEETPACK is subdivided into seven major sections or sets of programs:

1. An interactive-mode and a batch-mode data communicator (INTERACTER and BATCHER) containing the calculation order finder.
2. The executive control programs (PEETPACKSUBS, EXECUTIVESUBS).
3. A library of unit model routines (MODELSUBS).
4. A library of thermodynamic property routines (THERMOSUBS) and one of physical property routines (PHYSUBS).
5. A data-echoer program to print-out the stored data (ECHOER).
6. A bank-file of thermodynamic data constants (PROPTS) and one of physical property data constants (PPTEXTRA).
7. Two sets of two MACRO's each (one set for interactive-mode control, PEETREADER and PEETUPDATER, and the other for batch-mode control, BATCHREAD and BATCHUPDATE). They control the sequencing of the above six sections and the assignment of the various peripherals (card-readers, line printers, data files and card punchers) which are used to communicate data and information between the different sections, and with the user.

The overall organisation of PEETPACK is illustrated in figure 1 and a typical case-run sequencing by a MACRO is shown in figure 2. Its major sections are described below.

The data communicators

The data communication section is one of the most important parts of a flowsheeting program from the user's point of view since it is the one he is directly in communication with and which he must learn to use easily to have access to the whole package. The first impression it makes on him will normally affect his subsequent attitude towards the program as a whole.

The necessity to provide the correct mode of data communication for different classes of users (undergraduates, research postgraduates and process engineers) cannot be over-stressed (7), however the advantages of providing the users with both modes of data input and with the possibility of mixing these input modes are also valuable as the computer installations in different universities and industries are used at different levels of work load (but the running of simulation cases in interactive mode has not been fully developed).

The data communicators in PEETPACK have been given particular attention to overcome the practical problems associated with fixed-format or single-mode data inputs. PEETPACK offers two separate data communicators, each designed to operate in its own mode and to accept from the user the minimum essential data in a very free numeric and alpha-numeric formats and in a flexible sequence. The extensive use of backstore files allows the user to input his data once only in either mode and update them subsequently, also in either mode, with the minimum number of instructions.

Both programs have been designed as separate free-standing programs each responding to a particular set of data input commands which it recognises and according to which it interprets the data following it. The need to interpret the data arises from the form in which the user is allowed to input them. A number of restrictions inherent in older or less developed programs have been waived. In PEETPACK units and streams may be given alphanumeric names and the components used in the simulation are mentioned using their common names. The sequencing of the input data in batch-mode is left to the user though certain information must be present (input) before other data can be read in. In the interactive-mode, the user is lead by the program through the whole data input to facilitate the sequencing task, but he may choose to input his data in his own sequence by exiting fairly soon from the data reader (MACRO PEETREADER) and entering the updater (PEETUPDATER) which does not contain the question-and-answer input facility (a direct entry to the updater is not permitted). In both modes a data input and a data updater are available. They are basically the same program except that, depending on the MACRO used, a switch may be set or suppressed and the program then either clears the storage areas and sets the default value data for new data inputs or reads the data from the previous input or simulation run to update them.

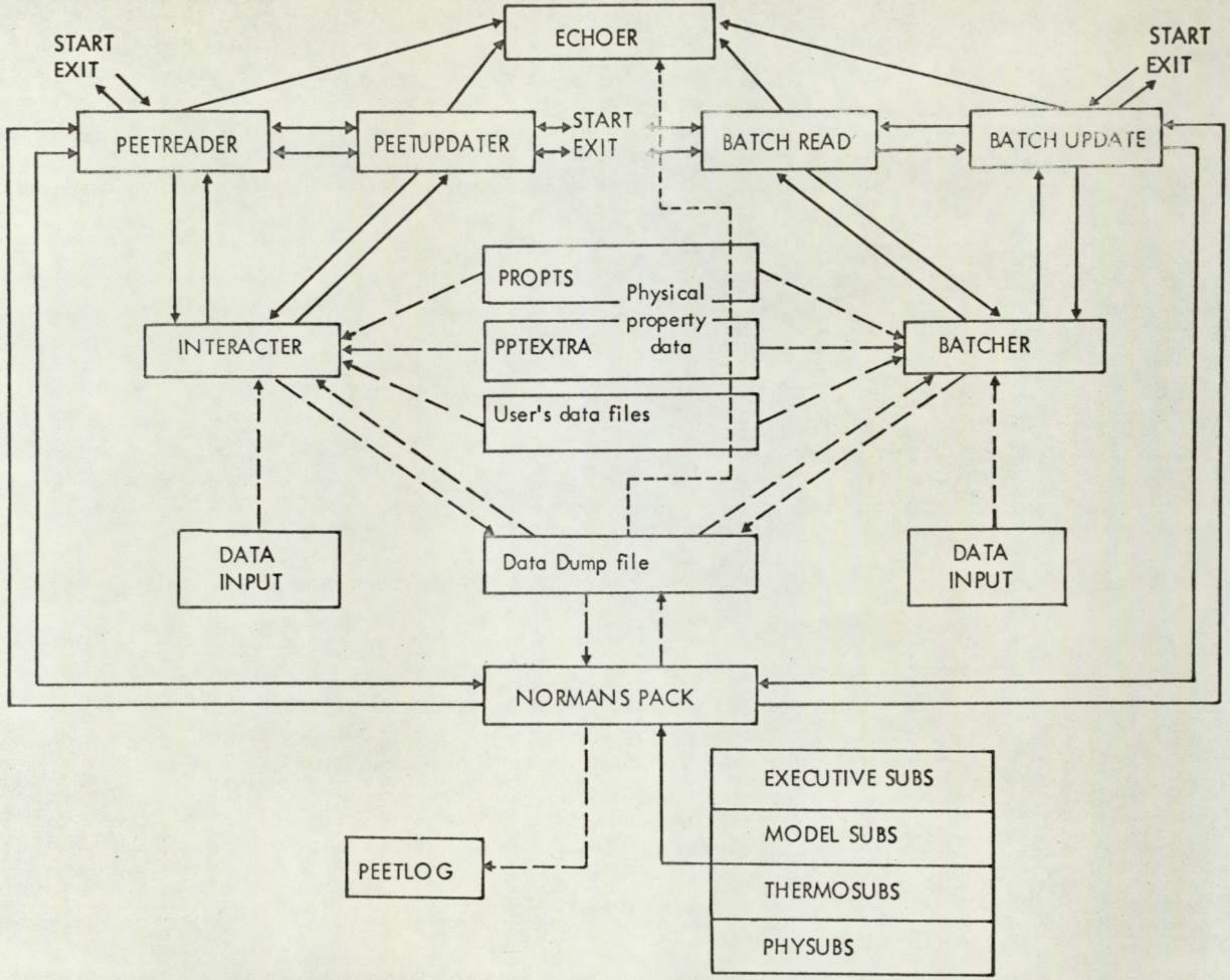
PEETPACK requests from the user the same type of data as do most flowsheeting programs; these are of two types:

Type 1 data: Process layout and operating conditions. These data include the process flow diagram (the PROCESS MATRIX), the operating conditions for each unit (the PARAMETER MATRIX), the physical description of the feed streams and possible iterate streams (the STREAMS MATRIX) and the name of the chemical compounds used and generated in the process (the COMPONENTS). These data are classified as Set-A Commands in figure 3.

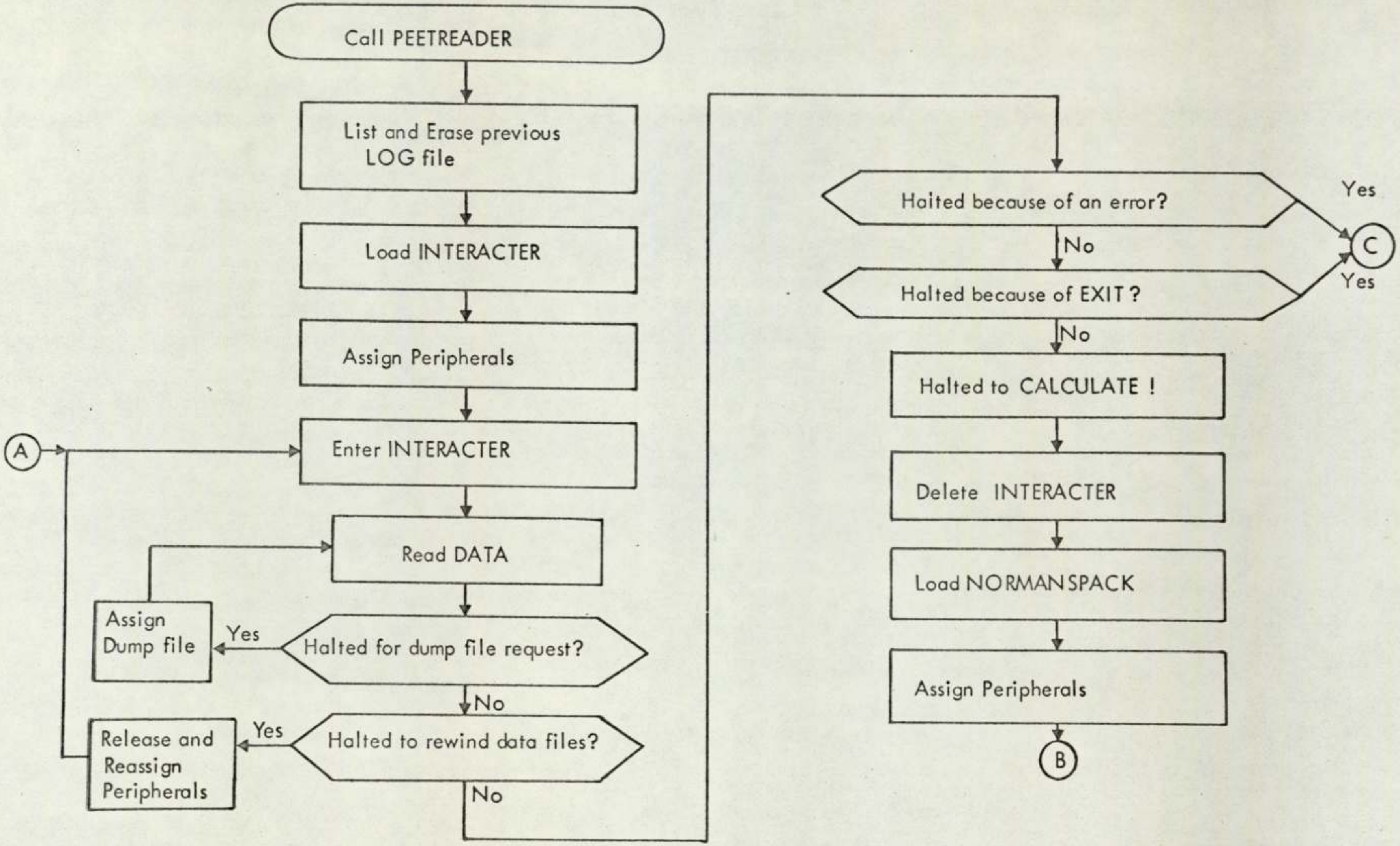
Type 2 data: Simulation control data (Sets B and C). The course of the simulation is controlled by the specification of the sequence in which the units are to be calculated (the ORDER OF CALCULATION). This order may be found automatically by the data communicator or set manually by the user. Model routine names must also be assigned to each unit (the ROUTINES NAMES). To check the convergence of recycle loop calculations and control the length of a run a TOLERANCE and the maximum number of LOOPS calculation must be set. Two ACCELERATORs are available to promote convergence and a flag (KEEPON/NOGO) decides upon the course of action to take should a loop fail to converge in "LOOPS" calculations. PEETPACK offers also a dual system of units for data communication between the user and the program, BRITISH UNITS and S.I.UNITS.

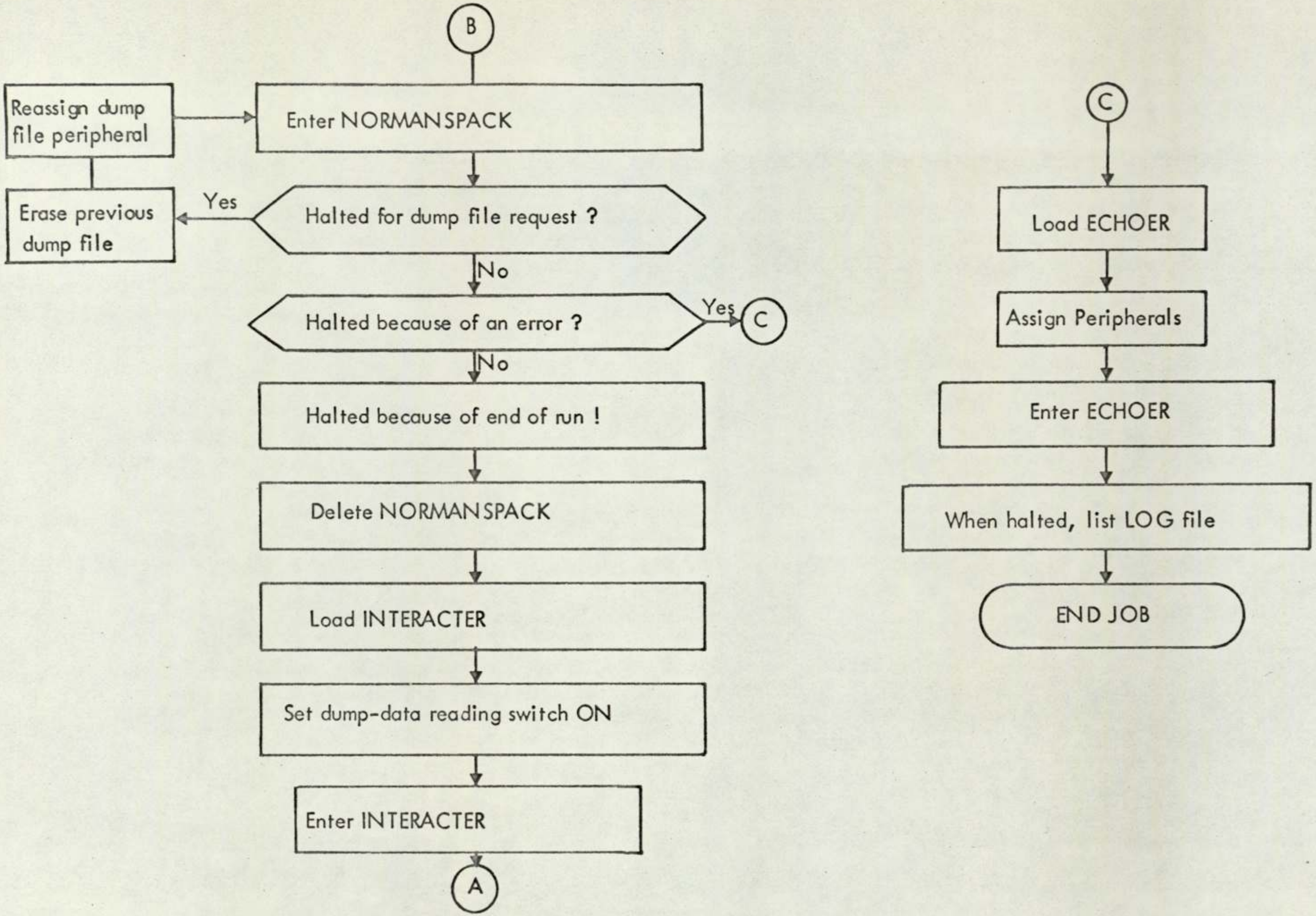
The names in capital letters are the commands understandable to the communicators. Most alphanumeric and numeric data may be punched anywhere on a card or on a teletype terminal line and a free-alphanumeric READ format specially created for PEETPACK locates and reads them.

At the end of a data input or update session the communicators dump the interpreted data into a dump file which is then accessible to the main flowsheeting program or to the data echoer, or again to the communicators.



ECHOER
START
EXIT
START
EXIT
PEETREADER
PEETUPDATER
START
EXIT
BATCH READ
BATCH UPDATE
PROPTS
Physical
property
data
PPTEXTRA
INTERACTER
BATCHER
User's data files
DATA
INPUT
Data Dump file
DATA
INPUT
NORMANS PACK
EXECUTIVE SUBS
MODEL SUBS
THERMOSUBS
PHYSUBS
PEETLOG


Call PEETREADER
List and Erase previous LOG file
Load INTERACTER
Assign Peripherals
A
Enter INTERACTER
Read DATA
Assign Dump file
Yes
Halted for dump file request?
No
Release and Reassign Peripherals
Yes
Halted to rewind data files?
No
Haited because of an error?
Yes
No
C
Halted because of EXIT?
Yes
No
Halted to CALCULATE !
Delete INTERACTER
Load NORMANSPACK
Assign Peripherals
B


B
Reassign dump file peripheral
Enter NORMANSPACK
Erase previous dump file
Yes
Halted for dump file request ?
No
Halted because of an error ?
Yes
C
No
Halted because of end of run !
Delete NORMANSPACK
Load INTERACTER
Set dump-data reading switch ON
Enter INTERACTER
A
C
Load ECHOER
Assign Peripherals
Enter ECHOER
When halted, list LOG file
END JOB

A - Section-Title Commands: (Batch-mode input)

PROCESS MATRIX
PARAMETER MATRIX
ROUTINES NAMES
COMPONENTS NAMES
LIBRARY COMPONENTS
ADDITIONAL COMPONENTS
NEW COMPONENTS
STREAM VALUES
PREFERRED STREAMS
ORDER OF CALCULATION
VARIABLES NAMES

A′ - Commands for data-section access and alteration:

```
First Part              Second Part                   Third Part

                     (  STREAM  --------------->  Stream name
                     (  UNIT         )
                     (  ROUTINE      )--------->  Unit name
                     (  PARAMETERS   )
INPUT   )            (                          (  STREAMS
ADD     )            (                          (  PARAMETERS
CHANGE  )----------> (  ALL  -----------------> (  ROUTINES
TYPE    )            (                          (  COMPONENTS
PRINT   )            (                          (  CONTROLS
                     (                          (  PREFERRED STREAMS
                     (  CONTROLS
                     (  COMPONENTS
                     (  ORDER OF CALCULATION
                     (  PREFERRED STREAMS
```

B - Data specification followed by data values:

	Default value
ACCELERATOR i	0
LOOPS l	0
TOLERANE t %	0
TRACE n	0

C - Commands requiring no additional data:

BRITISH UNITS
S.I. UNITS
NOACCELERATOR
NOTRACE
NOORDER OF CALCULATION
DATA ECHO
KEEPON
NOGO
CALCULATE
EXIT
END

Figure 3 : Data input commands

The calculation order finder

The optional calculation order finder is an integral part of the data communicators because of the facility given to the user to suggest preferred iterate streams, have the program check them for sufficiency to cut all loops and obtain the calculation order which he may then accept or reject. In the latter case he may propose new iterate streams and repeat the procedure.

The order finder is written in the classical three-part form:

1. The location and extraction of all elementary recycle loops (or partitioning).

2. The search for the optimum iterate streams to cut all recycle loops open (or tearing).

3. The ordering of the units in a calculable sequence (or sequencing).

The definition of "optimum iterate streams" is the accepted one of "the minimum number of iterate streams, including those chosen by the user, needed to tear all elementary loops open"; and by elementary loops it is meant loops involving units at most once in their complete sequence (11).

1. Partitioning

PEETPACK uses the exhaustive path searching method to locate all the elementary loops in a process flowsheet. This method is much more efficient that the adjacency matrix method (10) in locating elementary loops whereas the matrix method is preferable in locating maximal cyclic nets (sets of nested loops having one or more units in common). The method used in PEETPACK is theoretically very efficient and locates all possible loops (11). It is based on the extension of paths formed from one node to another following in turn each output from the first node as far as it can go through other units until it either returns to the initial node (hence locating an elementary loop) or until a plant product stream is met (a partly or completely sequential path). Once a path (sequential or recycle) is terminated, the program looks at the one before last unit in the sequence and tries to follow its other output streams in order to locate other paths. Every path is traced forward then backwards to locate all branches in the loop and trace them all out. Once all loops in which a particular unit is involved are located, the next unit in the plant is checked out identically until the last unit is reached.

2. Tearing

Locating the optimum iterate streams has followed largely the approach of Lee and Christensen (12). In their method they create a boolean matrix where rows represent the elementary loops in a graph and the columns represent the streams. Non-zero elements in the matrix show the streams involved in each loop. Each elementary loop requires only one torn stream to be opened, but since many loops may have one or more streams in common, these are the more suitable ones to cut. The matrix is used to locate streams appearing in successively larger number of loops. If all other streams in the loops in which each observed stream appears have not yet been deleted, the observed stream itself is deleted by setting all its non-zero elements in the matrix to zero. By successively eliminating the streams appearing in the least number of loops, the optimum few remain as iterate streams.

PEETPACK uses the same method but incorporates the added facility of letting the user set his preferred iterate streams and by so doing overwrites the procedure's option of deleting these streams unless one or more of them is unnecessary as an iterate stream. It sets the number of loops the preferred streams appear in to a higher number than that of the other streams (i.e. to the number of elementary loops + 1) so that they are checked last. This ensures that the minimum number of iterate streams, including the preferred ones remain. In the interactive data input mode, these iterate streams are presented to the user who can either accept them or reject them and propose a different choice of iterate streams, and the above procedure is repeated. A similar procedure was prepared by Forder and Hutchison (13) for CONCEPT.

3. Sequencing

Plant feed streams and iterate streams are flagged as "known" streams and given a flag of 1. The elementary loops which were used to locate the iterate streams are merged together to form maximal cyclic nets since it is more efficient to converge the calculation of maximal nets rather than individual elementary loops. The maximal nets matrix is converted to show the units making up the loops rather than the streams. The program searches for the unit whose inputs all bear a flag of 1. It checks whether it is part of a loop. Units in sequential sets are flagged 1 immediately, added to the calculation order and their output streams flagged likewise. For a unit within a recycle loop, the maximal cyclic loop in which it appears is treated itself as a sub-system. The unit under consideration is set in the sequence and is flagged along with its output streams with a temporary flag 3. All the units in the loop are checked in the hope of adding them all to the calculation sequence. If such is the case all temporary streams are set to permanent ones and the next unflagged unit in the plant is checked. Otherwise all temporary flags are set back to zero and the added recycle units deleted from the sequence. The scanning of the main list of units is repeated until all units are flagged thus providing the complete calculation order.

The executive program

The executive program in PEETPACK controls only the simulation calculations and accelerates their convergence as it receives the calculation order and the data already processed from the data communicators through the dump file. The executive's functions have been purposely reduced to contain its size and running time.

The executive programme is loaded into core by the MACRO following the input or update of the simulation data. It reads the interpreted data from the dump file and calls the computation controller routine. This routine analyses the calculation order and calls each unit model in the predetermined order after loading its input streams descriptions and its parameters in matrices accessible to the models. The models may be called either by a fixed model-calling routine which is part of the executive or by another version of this routine created by the communicators. The first alternative is used when PEETPACK is controlled by one of the four standard MACRO's mentioned earlier, and would be the version commonly used when the whole package is easily available to the users, such as in academic institutions. It is assumed here that a user will not need to add any other models during a run or that he will give his models certain fixed names and recompile the executive with these routines before starting his run. The second alternative is offered to industrialists who may not have easy access to the executive and who wish to incorporate their own models and give them their own names even if these already exist in PEETPACK's model library. A unit calling sequence is generated by the communicators and appended to the executive which is then recompiled in the presence of the user's model library, hence waiving any restrictions or obstacles to the ease and versatility of the whole package.

Upon return from the model calculations, the controller checks whether it is part of a recycle loop and if so checks the convergence of its output streams descriptions before re-storing them in the main storage matrices. The convergence of recycle loops may be optionally accelerated by either a Wegstein-type convergence promoter (14) or an Orbach-type promoter (10), the first one being of general use and the second one more suitable to systems made-up of linear models (such as sets of splitters and mixers). Recycle loops are calculated repetitively until convergence is achieved or the maximum number of loop calculations set by the user is exceeded in which case the run may either stop or proceed to the next sequence of units, depending on a user-set switch (NOGO/KEEPON).

At the end of a run, the final results are printed on the line printer and a copy is sent to the dump file to be used by the data communicators for updating or by the data echoer at the end of a simulation.

The library of unit models

The unit operation models and the thermodynamic and physical data evaluation routines are the main features in a package on which is based its wide acceptance or its failure at the industrial level, but are of lesser importance as teaching tools (7). However the absence of general data routines complicates tremendously the preparation of unit models.

To overcome these problems, a number of unit operation models and property evaluation routines are offered in PEETPACK.

The unit operation models are mainly heat and mass balancing routines operating on input stream information to produce the output stream descriptions subject to operating conditions or constraints imposed in the models' parameter lists. The models emphasise the heat and mass balances rather than the mechanical design aspect. This approach was entirely due to time limitations and it is hoped additional models will be written and freely exchanged between users.

The unit operation modelled in PEETPACK are as follows:

1. A flow only mixer and new composition setter (routine MASSMIX) useful for mass balance only problems.
2. A general heat and mass mixer (MIXER).
3. A stream divider into many outputs (SPLITTER).
4. A combined mixer then divider (SPLITMIX).
5. An adiabatic throttle valve or pressure setter (VALVE).
6. An adiabatic flash column (FLASHER).
7. A stream purifier or components remover (PURIFYER).
8. An isothermal pump or pressure setter (PUMP).
9. A polytropic gas compressor (COMPRESR).
10. A stoichiometric adiabatic reactor (REACTOR) to which can be supplied the heat of reaction and an extent of reaction factor.
11. A multiphase general counter-current heat exchanger (HEATEXCH).
12. A heater/cooler or pressure setter (HEATER).

13. A general, multicomponent, tray to tray, distillation column model with optional reboiler model attached (DISCOL).

14, 15. A stream bubble point and dew point setters (SETBP, SETDP). All the models rely entirely on the property package, described below, for their thermo-physical data and their accuracy depends to a large extent on these data evaluation routines. The writing of user-provided models can also be simplified by using these routines. An idea of the scope of the models may be obtained from table 1 which illustrates the parameters required to characterise the models.

<u>The thermodynamic and physical data evaluation routines</u>

These essential complements to the unit models library are set in two distinctive libraries. The first one contains the thermodynamic and thermal data evaluation routines. These are the more important of the two sets during the heat and mass balance calculations and are sufficient to help specify completely the state of pure components and mixtures in either the vapour, the liquid or the two-phase region. The data evaluations have been divided into a number of routines each fulfilling one function only. The methods chosen in these routines are the best known and most tested ones, and their incorporation in the best packages (8) is a sufficient guarantee of their worth. These routines are:

1. KVALUE: calculates the vapour-liquid equlibrium constants using the Chao-Seader method (15) for non-polar organic compounds and light gases, which are the only compounds catered for, yet, in PEETPACK.

2. EVALUE: is a very simple routine provided for the users who wish to input their own correlations of K-values as functions of temperature.

3. ENTHALPY: calculates the enthalpy of organic compounds and gases obeying the corresponding-state principle using the generalised equations compiled by Yen and Alexander (16).

4. TEMP: calculates the temperature and vapour fraction of a stream based on its enthalpy, pressure and composition by an interpolation search technique (Reguli-Falsi).

5. DEWBUB: calculates the dew point or bubble point of a mixture by forcing the equilibrium composition of the liquid and the vapour phase to sum up to unity at these temperatures.

6. FLASH: separates isothermally a two-phase mixture into its pure phases fractions.

7. VAPRES and BOILPT: calculate the vapour pressure and boiling point of pure components using an Antoine-type equation (17).

The second routine library contains physical property evaluation data usually used in mechanical design routines. They use the most accurate general methods suggested by Reid and Sherwood (9), unless otherwise specified, and may be easily extended to cater for many compounds of different types. These routines are as follows:

1. DENSTY: The density of gaseous mixtures are calculated using the Reddlich-Kwong equation of state (18). The liquid mixture density is calculated using either special forms of curve-fitted correlations or compressibility factors taking into account pressure and temperature effects on base point densities (19). Two-phase mixtures combine both methods according to a mixing rule based on the addition of both phase volumes.

MODEL	Parameter position	Description
MASSMIX	-	None
MIXER	-	None
SPLITTER	1 to no. of outputs	Fractional Splits
SPLITMIX	1 to no. of outputs	Fractional Splits
VALVE	1	Output Pressure
FLASHER	1	Output Pressure
PURIFYER	1	Number of components to remove (N)
	2 to N+1	Components positions in the Component order read in
PUMP	1	Output Pressure
	2	Mechanical Efficiency (Optional)
COMPRESR	1	Output Pressure
	2	Isotropic Efficiency
	3	Mechanical Efficiency
	4	Ratio of C_p/C_v
REACTOR	1 to no. of components	The Reaction Stoichiometric Coefficients
	N+1	Extent of Reaction (optional)
	N+2	Heat of Reaction (optional)
HEATEXCH	1	Temperature of first outlet stream
	2	Exchanger's Surface Area
	3	Heat Transfer Coefficient
	4	Pressure Drop in first stream
	5	Pressure Drop in second stream
	6	Set to 1 if parameter 1 represent difference between T out 1 and T in 2 otherwise set to 0
		(use 1 or 2 & 3, set unused onesto 0)
HEATER	1	Output Temperature
	2	Pressure Drop in the stream

DISCOL	1	Number of Plates in Column
	2	Feed Plate
	3	Tops Product Flow rate
	4	Vapour Flow rate in the Top Product
	5	Reflux Ratio
	6	Column Pressure
	7	Light Key Component number
	8	Heavy Key Component number (as in the order read in)
	9	Pressure Drop in the stream heating the reboiler (optional, default = 0.0)
	10	Number of terms in the K-value vs temp. Correlation (optional)
	11	Temperature code used = 1 for degrees F 2 for degrees C 3 for degrees R 4 for degrees K
	12 onwards	The coefficients for the K-value correlations starting with the first component in the list. The coefficients are read in as: A, B x 1.0E2, C x 1.0E4, D x 1.0E7, E x 1.0E10, F x 1.0E12, G x 1.0E14 for the equation: $K = A + B \times T + C \times T^2 + D \times T^3 + E \times T^4 + F \times T^5 + G \times T^6$ (parameters 9 onwards are optional)
SETDP	-	None
SETBP	-	None

Table 1 : The Parameters for the PEETPACK models

2. SPHEAT: The specific heat of gaseous hydrocarbons and light gases is calculated by the corresponding-state-approach method of Hirschfelder et al (19, 20). The liquid phase specific heats are calculated from polynomials fitted to data obtained from the most accurate and consistent sources (21, 22). The specific heat of two-phase mixtures is not calculated.

3. VISC: The vapour phase viscosity is calculated using the Chapman-Enskog-Cowling equation (19) and the viscosity of the liquid phase viscosities are calculated by one of the following methods. A polynomial fit as function of temperature (21) for accurate results or either Souder's method or Thomas's method, or Stiel and Thodos's method, whichever is more accurate for each individual component if polynomials are not available. Two-phase mixture viscosities are set equal to the liquid phase viscosity.

4. CNDVTY: The thermal conductivity of gases is calculated by the Misic and Thodos method, but only for non-polar gases. For liquids the Robbins and Kingrea method is used. The conductivity of two-phase mixtures is set equal to the liquid phase conductivity. Polar compounds are not considered.

5. SURFTENS: The surface tension of non-polar liquids is computed by the Macleod-Sugden method. For polar compounds no reliable method is available and the same one is used.

Besides being offered as part of PEETPACK, these thermodynamic and physical property evaluation routines have been incorporated in a separate free-standing package for property evaluation: PROPACK (Property Package) which may be used either interactively or in batch mode. These routines require a number of constants and coefficients and these are set in two property data files which contain information for 45 hydrocarbons and light gases. More data may be input to PEETPACK either in different property files or as continuation to the same files.

Conclusion

PEETPACK has been used successfully to solve a number of simulation problems (23), and its thermodynamic and physical property evaluation package was used in the design of a number of unit operation equipments (24). Its main drawback is that it uses a number of features offered by the ICL 1900-series computers operating with the GEORGE 3 or 4 system. Its conversion to fit other operating systems or computers is possible but time consuming. This strong dependence on operating systems is unavoidable because of the special alphanumeric character handling facilities needed in the data communicators, the need for MACRO's (Job Control language commands) to control the operation of the package as a whole, and because of the special facilities used to trap errors and avoid the loss of information in unsuccessful runs.

PEETPACK and PROPACK are available on request from the authors on payment of a nominal handling charge.

Bibliography

1. Flower J.R. and Whitehead B.D.; The Chem. Engr., No. 273 (1973), P271.

2. GEMCS; available from Professor A.I. Johnson, Dean of Engineering Sciences, University of Western Ontario, London, Ontario, Canada.

3. ESSPROSS; available from J.W. Ponton, School of Engineering Sciences, U. of Edinburgh, Kings Buildings, Mayfield Road, Edinburgh 9.

4. UNICORN; available from G.J. Forder, Systems Engineering Group, Techn. Chem. Labor , E.T.H., Zürich, Switzerland.

5. CONCEPT; Chemical Engineering Group, The Computer-Aided Design Centre, Madingley Road, Cambridge, CB3 OHB, U.K.

6. FLOWPACK; I.C.I., Corporate Laboratories, P.O. Box 13, The Heath, Runcorn, Cheshire.

7. Peters N. and Barker P.E., The Chem. Engr., London, No. 283, March 1974.

8. Worley R.L. and Motard R.L.; "Chemical Engineering Computing, Vol. 2", P4, A.I.Ch.E., New York 1972.

9. Reid R.C. and Sherwood T.K., "The properties of gases and liquids", McGraw-Hill, New York 1966.

10. Crowe C.M. et al; "Chemical Plant Simulation", Prentice Hall, New York 1971.

11. Tiernan J.C.; Comm. A.C.M., 13 (1970), ≠ 12, P722.

12. Lee W. and Rudd D.F.; A.I.Ch.E.J., 12 (1966), ≠ 6, P1184.

13. Forder G.J. and Hutchison H.P.; Chem. Eng. Sci., 24 (1969), P771.

14. Wegstein J.H.; Comm. A.C.M., 1 (1958), ≠ 6, P9.

15. Chao K.C. and Seader J.D.; A.I.Ch.E.J., 7 (1961), ≠ 4, P598.

16. Yen L.C. and Alexander R.E.; A.I.Ch.E.J., 11 (1965), ≠ 2, P334.

17. Handbook of Chemistry and Physics; 54th Edition, PD-162, C.R.C. Press, Cleveland 1973.

18. Reddlich O. and Kwong J.N.S.; Chem. Review, 44 (1949), P233.

19. American Petroleum Institute : Technical data book, A.P.I. Division of Refining, New York 1966.

20. Hirschfelder J.O. et al; Ind. Eng. Chem., 50 (1958), P386, Ind. Eng. Chem. Fundam. 1 (1962), P224.

21. Gallant R.W.; Hydrc. Process, 49 (1970), ≠ 4, P132.

22. Timmermans J.; "Physico-chemical constants of pure organic compounds", Elsevier Co., New York 1950.

23. Peters N.: PhD Dissertation, 1974, Dept. of Chemical Engineering, The University of Aston, Birmingham 4, U.K.

24. Kilikas A.C.; MSc Dissertation, 1973, Dept. of Chemical Engineering, The University of Aston, Birmingham 4, U.K.

Plant data reconciliation by flowsheeting

By N. Peters and Professor P.E. Barker
Department of Chemical Engineering,
The University of Aston in Birmingham.

Summary

The checking and reconciliation of heat and mass flow rates from plant survey reports, and the estimation of a complete set of heat and mass flows from a few inaccurate or inconsistent spot measurements are two ideas that have been developed and incorporated in Aston University's new and non-proprietary flowsheeting program PEETPACK (Process Engineering Evaluation Techniques Package). This new application recreates a complete heat and mass model of the plant from the description of any number of the plant streams, each, including the feed streams, subject to a known average range of deviation from its true value. It bases the data reconciliation on the minimisation of the sum of the squares of the deviations between the given meter readings and their calculated values.

Flowsheeting programs have been found particularly useful to this application in that they minimise the amount of work and data a model manipulation previously required to reconcile the given data. A library of models can be called upon to calculate the unmeasured streams or unknown variables, or in the abscence of sufficiently reliable models, guesses may be input and given very low weighting factors to avoid upsetting the overall balances.

In this paper, four of the better known reconciliation methods are analysed from the point of view of their data requirements and of the special model and equation preparations they expect from the user. The more suitable one for use in a flowsheeting program is then presented in more detail and its implementation in PEETPACK is illustrated through the solution of an industrial-type problem.

APPENDICES

APPENDIX ONE .

Contents :

Code Sheets for data input to the Birmingham University version of PACER.

Input data to PACER (Aston University version) for the Ethane/Ethylene separation plant.

Input data to GEMCS for the Ethane/Ethylene separation plant.

PACER CODING SHEETS

To facilitate the organization of data for a case study of an existing or proposed plant operation, coding sheets have been prepared. These require some care in their use:-

i) integer numbers must be used as required by the name of the variable, e.g. KRUN, N2MAX, NOEN, etc. are all integer variables.

ii) integer variables must be "right justified" in the fields indicated by the numbers along the tops of the tables of the coding sheets.

iii) floating point or decimal numbers may be placed anywhere within the fields indicated. The F notation must not be used.

The following is the order of data:-

1. Coding Sheet A

a) Set up size of DIMENSION ARRAYS IN PACER (two cards)
(NE) Specify number of equipments to be used (one card)
i.e. number of rows in MATRIX

ENAME ( ) Give order of names of equipments used as defined by MATRIX. Name of an equipment may be defined by 6 alphanumeric characters (one card).

NOTE: PACER TESTER DOES NOT REQUIRE LAST TWO CARDS

b) Run data (one card)

KRUN is the run number
KSETS is the printing control number
=0: for initial conditions, intermediate printing every KPRINT loops, and final answers (normal mode)
=1: for printing in equipment subroutines for trace of equipment calculations
=2: for trace of equipment subroutines and in addition will print STRM for each equipment and (in TEST) the number of deviations from test vector values
=3: in addition to the above will give flag lists

It is apparent that in large problems, any value of KSETS other than zero will give a very large amount of printing and should never be used on teletype.

KCLEAN when zero ensures that matrices are initially zeroed
LOOPS is the maximum number of iterative loops to be allowed in a recycle calculation
NOGO if zero, will use results to do a further calculation; if a positive number will go on to the next case
KARDS if positive causes the stream variables matrix to be punched at the conclusion of a run whether or not convergence obtained
KPRINT is the number of iterations between printout of the stream variables matrix and is automatically made 5 unless otherwise specified
KONV is a number to control attempts to accelerate the convergence after KONV loops.

c) Dimension data (one card)

On this card are:-

N2MAX - number of rows in the process matrix
N3MAX - number of columns in the process matrix (1 + maximum of st
NEMAX - maximum equipment number(not necessarily max. number of equipments)
NSMAX - maximum stream number (not necessarily max. number of streams)
NELMAX - length of equipment parameter vector
NECLMX - length of equipment control vector
NSLMAX - length of stream variables vector
NSCLMX - length of stream properties vector
NPSMAX - maximum number of preferred stream numbers (not greater than 3)
NOCOMP - number of components

2. Coding Sheet B (Process Matrix Sheet)

NOKPM is the number of rows of the KPM matrix to be read in. For each row there will be:-

i) an equipment number corresponding to the information flow diagram

ii) an equipment subroutine name, found in the library

iii) a code name, corresponding to plant flow sheets

iv) now have the streams associated with the equipment number, input streams followed by output streams, the latter made negative. Any rules for ordering the input and output streams set up by the equipment subroutine must be followed.

3. Coding Sheets C1 and C2 (Equipment and Stream Information)

The order expected by the executive program is:-

Sheet C1

a) NOEN is the number of equipment vectors to be read; this is followed by the equipment vectors themselves, each of which must have a length NELMAX.

b) NOENC is the number of equipment control vectors;this is followed by the equipment control vectors, each of which must have a length NECLMX.

Sheet C2

c) NOSN us the number of stream variables vectors to be read; this is followed by the stream variables vectors, each of which has a length NSLMAX.

d) NOSNC is the number of stream properties vectors to be read; this is followed by the stream properties vectors each of which has a length NSCLMX.

If NOENC and NOSNC are zero, the program will automatically move to the next part of the data to be read in; however, cards (which may be blank or zero) must be in the data for these numbers.

4. Coding Sheet D (Test vectors)

If NDELS is a positive number, the following test vector, having NSLMAX numbers will be read in.

NPS is the number of preferred streams to be read (not larger than 3); followed by the preferred steeams.

5. Coding Sheet E (Additional equipment vectors)

NOAEN is the number of additional equipment vectors to be read in, with a maximum of 4; followed by the additional equipment vectors, each of which must have a maximum length of 40 at present.

6. Coding Sheet F (Calculation Order List)

a) If KK is greater than zero a calculation order list is to be read in; this list allows bypassing of GUESS1 and GUESS2 in the executive program and avoids appreciable computation time.

LISTT is the number of equipment units in the NELIST list which follows.

In the NELIST the first word is the equipment number, the second word is,

1 for direct calculation
2 for equipment in an iterative calculation
3 is the last equipment in an iterative calculation.

b) KSPRINT is the number of stream vectors to be printed each loop, followed by the numbers of the streams to be printed (with a maximum of 9 streams).

[redacted]

PACER CODING SHEET A

Set Up Dimension of Pacer

2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64	66	68	70	72	74
VA	VB	VC	VD	VE	VF	VG	VH	VI	VJ	VK	VL	VM	VN	VO	VP	VQ	VR	VS	VT	VU	VX	VY	VZ	UA	UB	UC	UD	UE	UF	UG	UH	UI	UJ	UK	UL	UM

UN	UO	UP	UQ	UR	US	UT	UU	UV	UW
76	78	80	2	4	6	8	10	12	15

Next card

5
NE

6	12	18	24	30	36	42	48	54	60	66	72	78
ENAME (1)	ENAME (2)	ENAME (3)	ENAME (4)	ENAME (5)	ENAME (6)	ENAME (7)	ENAME (8)	ENAME (9)	ENAME (10)	ENAME (11)	ENAME(12)	ENAME (13)

5	10	15	20	25	30	35	40	45	50
KRUN	KSETS	KCLEAN	LOOPS	NOGO	KARDS	KPRINT	KONV		
N2MAX	N3MAX	NEMAX	NSMAX	NELMAX	NECMAX	NSLMAX	NSCLMX	NPSMAX	NOCOMP

[redacted]

PACER CODING SHEET B

Coded by ______________ Run ______________ Date ______________

Checked by ______________

Process Matrix

Page ______________ of ______________

5 NOKPM	7 — 12	14 — 19	25	30	35	40	45	50	55	60	65

PACER CODING SHEET C1

Equipment Vectors

EN

5	15	30	45	60	75

ENC

5	15	30	45	60	75

PACER CODING SHEET C2

Stream Vectors

SN

5	15	30	45	60	75

SNC

5	15	30	45	60	75

PACER TESTER ENDS WITH THIS SHEET

PACER CODING SHEET D

ded by ____________ Test Vectors Run ____________ Date ______ ______

ecked by ____________ Page ____________ of ______ ______

5

NDELS

	15	30	45	60	75

PREFERRED STREAM NUMBERS

5

NPS

	10	15

Special Instructions

Coded by ______________

Checked by ______________

NOT USED FOR PACER TESTER

PACER CODING SHEET E

Additional Equipment Vectors

Run ______________ Date ______________

Page ______________ of ______________

5

15	30	45	60	75

15	30	45	60	75

NOT USED FOR TAPER TESTER

Coded by ______________

Checked by ______________

PAPER CODING SHEET F

Calculation Order List

Run ____________ Date ______________

Page ____________ of ______________

KK 5	LIST 10	15	20	25	30	35	40	45	50

KSPRNT

5	10	15	20	25	30	35	40	45	50

Input data to PACER

(Models Used)

(Dimensions and Controls)

(Process Matrix)

(Parameter Matrix)

```
 13
ROTLDISCOLHEXCH COMPRSTHROTL
LITRCOOLERTHROTLTHROTLMIXERS
LITRTHROTLTHROTL
HANE / ETHYLENE  PROBLEM ON  PACER
  1    3    0  100    0    0    1    5
 13   20   13   19   14    2   11    2    2    2
 13
  1 THROTL              1   -2    0    0    0    0    0    0    0
  2 DISCOL              2   16    7   -3   -8  -18    0    0    0
  3 HEXCH               3   13   -4  -14    0    0    0    0    0
  4 COMPRS              4   -5    0    0    0    0    0    0    0
  5 THROTL              5   -6    0    0    0    0    0    0    0
  6 SPLITR              6   -7  -10    0    0    0    0    0    0
  7 COOLER             10  -11    0    0    0    0    0    0    0
  8 THROTL             11  -12    0    0    0    0    0    0    0
  9 THROTL              8   -9    0    0    0    0    0    0    0
 10 MIXERS              9   12  -19    0    0    0    0    0    0
 11 SPLITR             14  -15  -17    0    0    0    0    0    0
 12 THROTL             15  -16    0    0    0    0    0    0    0
 13 THROTL  THROTL     19  -13    0    0    0    0    0    0    0
 13
             0.           0.           215.          0.
             0.           0.           0.            0.
             0.           0.           0.
             0.           2.           40.           20.
5.           230.         230.         0.            4.6
5            .05          1.           2.
             0.           0.           452.          0.
             0.           0.           0.            0.
             0.           0.           0.
             0.           0.           600.          .75
4            1.23         0.           0.            0.
             0.           0.           0.
             0.           0.           457.          0.
             0.           0.           0.            0.
             0.           0.           0.
             0.           0.           .9            .1
             0.           0.           0.            0.
             0.           0.           0.
             0.           0.           439.          0.
             0.           0.           0.            0.
             0.           0.           0.
             0.           0.           414.          0.
             0.           0.           0.            0.
             0.           0.           0.
             0.           0.           414.          0.
             0.           0.           0.            0.
             0.           0.           0.
.            0.           0.           0.            0.
             0.           0.           0.            0.
             0.           0.           0.
```

.	0.	0.	.82	.18	
	0.	0.	0.	0.	
	0.	0.	0.		
.	0.	0.	215.	0.	
	0.	0.	0.	0.	
	0.	0.	0.		
.	0.	0.	402.	0.	
	0.	0.	0.	0.	
	0.	0.	0.		
0					
19					
	1.	511.	440.	345.	
25.	0.45	0.55	0.	0.	(Stream
	0.	0.	0.	0.	Matrix)
	0.	0.	0.	0.	
	0.	0.	0.	0.	
	0.	0.	0.	0.	
	0.	1325.	452.	215.	
00.	0.95	0.-5	0.	0.	
	0.	0.	0.	0.	
	0.	0.	0.	0.	
	0.	0.	0.	0.	
	0.	0.	0.	0.	
	0.	0.	0.	0.	
	0.	0.	0.	0.	
	0.	0.	0.	0.	
	0.	0.	0.	0.	
	0.	0.	0.	0.	
	0.	0.	0.	0.	
.	0.	0.	0.	0.	
	0.	0.	0.	0.	
.	0.	0.	0.	0.	
	0.	0.	0.	0.	
.	0.	0.	0.	0.	
	0.	0.	0.	0.	
.	0.	0.	0.	0.	
	0.	0.	0.	0.	
.	0.	0.	0.	0.	
	0.	0.	0.	0.	
.	0.	0.	0.	0.	
	0.	0.	0.	0.	
.	0.	1090.	413.	215.	
00.	0.95	0.05	1.	0.	
.	2.	0.	0.	0.	
	0.	0.	0.	0.	

```
.          2.            0.            0.            0.
           0.            0.            0.            0.

.          0.            0.            0.            0.
           0.            0.            0.            0.
  0
 11
0.001        0.001         0.001         0.001         0.001  (Toleran-
0.001        0.001         0.001         0.001         0.001
0.001                                                          ces)
  2
  4    16                                          (Preferred Streams)
  0
 13    13
  1     1     4     2     5     2     6     2     7     2
  8     2     2     2     9     2    10     2    13     2   (Calculation Order)
  3     2    11     2    12     3
  0
**
```

Input data to GEMCS

HANE / ETHYLENE PLANT ON GEMCS					(Title)
	1.	1.	1.	0.	
.	2.				(Print Switches)
	4.	5.	6.	7.	
	2.	9.	10.	14.	
	11.	12.	13.		(Calculation Order)
.					
	2.	2.	2.	7.	
	7.	2.	2.	2.	
	2.	2.	2.	2.	(Stream Codes)
	6.	6.	2.		
	0.	511.	440.	345.	
25.	.45	.55	1.	0.	(Feed and Guessed Streams)
	0.	1800.	452.	215.	
00.	.95	.05	0.	0.	
.	0.	1085.	413.	215.	
00.	.95	.05	1.	0.	
.					
	1.	16.	0.	0.	
	1.	0.	0.	0.	
	2.	0.	0.	0.	
5.					
	2.	26.	0.	0.	(Parameter Matrix)
	2.	16.	7.	0.	
	3.	8.	18.	0.	
.	20.	215.	230.	230.	
0.	.95	.05	1.	2.	
	3.	16.	0.	0.	
	3.	13.	0.	0.	
	4.	14.	0.	0.	
0.					
	4.	20.	0.	0.	
	4.	0.	0.	0.	
	5.	0.	0.	0.	
5.	.75	.94	1.23	0.	
	1.	16.	0.	0.	
	5.	0.	0.	0.	
	6.	0.	0.	0.	
7.					
	5.	17.	0.	0.	
	6.	0.	0.	0.	
	7.	10.	0.	0.	
2	.08				
	6.	17.	0.	0.	
	10.	0.	0.	0.	
	11.	0.	0.	0.	
9.	0.				
	1.	16.	0.	0.	
	11.	0.	0.	0.	
	12.	0.	0.	0.	
4.					

	1.	16.	0.	0.
	8.	0.	0.	0.
	9.	0.	0.	0.
4.				
	7.	15.	0.	0.
	9.	12.	0.	0.
	9.	0.	0.	0.
	5.	17.	0.	0.
	14.	0.	0.	0.
	15.	17.	0.	0.
2	.18			
	1.	16.	0.	0.
	15.	0.	0.	0.
	16.	0.	0.	0.
5.				
	8.	32.	0.	1.
	0.	0.	0.	0.
	0.	0.	0.	0.
	20.	4.	0.	1.
	1.	999.	0.	0.
	0.	0.	0.	0.
	0.			
	1.	16.	0.	0.
	19.	0.	0.	0.
	13.	0.	0.	0.
2.				
**				

APPENDIX TWO .

Contents :

Description of the Hooke and Jeeves optimisation routine added to the GEMCS executive and used to solve the second example in Chapter 3.

Flowchart of the optimisation routine.

Listing of the optimisation routine.

Description :

The variables in the equipment vector are renamed for convenience. Each variable is increased by its step length, one at a time, and all constrains are checked. If they are not violated the design variables in the appropriate unit vectors are updated and the calculation of the objective function proceeds. When a constraint is violated, a step in the other direction is taken and the constraints are rechecked. Another violation raises the failure index by one and the variable is reset to its base value. When a better objective function value is reached the stored value is updated. When all the variables have been increased or decreased, a pattern move is made from the new point in a direction and by a length equal to the overall move from the last base point. A successful move sets a new base point; an unsuccessful one triggers again the individual search in all directions from the new point reached. A failure in all directions means that the optimum is close and the step sizes are halved. The overall procedure is repeated until the step lengths are reduced below a certain size, then the search ends and the last values represent the optimum values.

The NCALL variable is used because the routine must be exited every time a move is made to calculate the objective function. For more details the Hooke and Jeeves optimisation methods should be studied.

The module's parameter array:

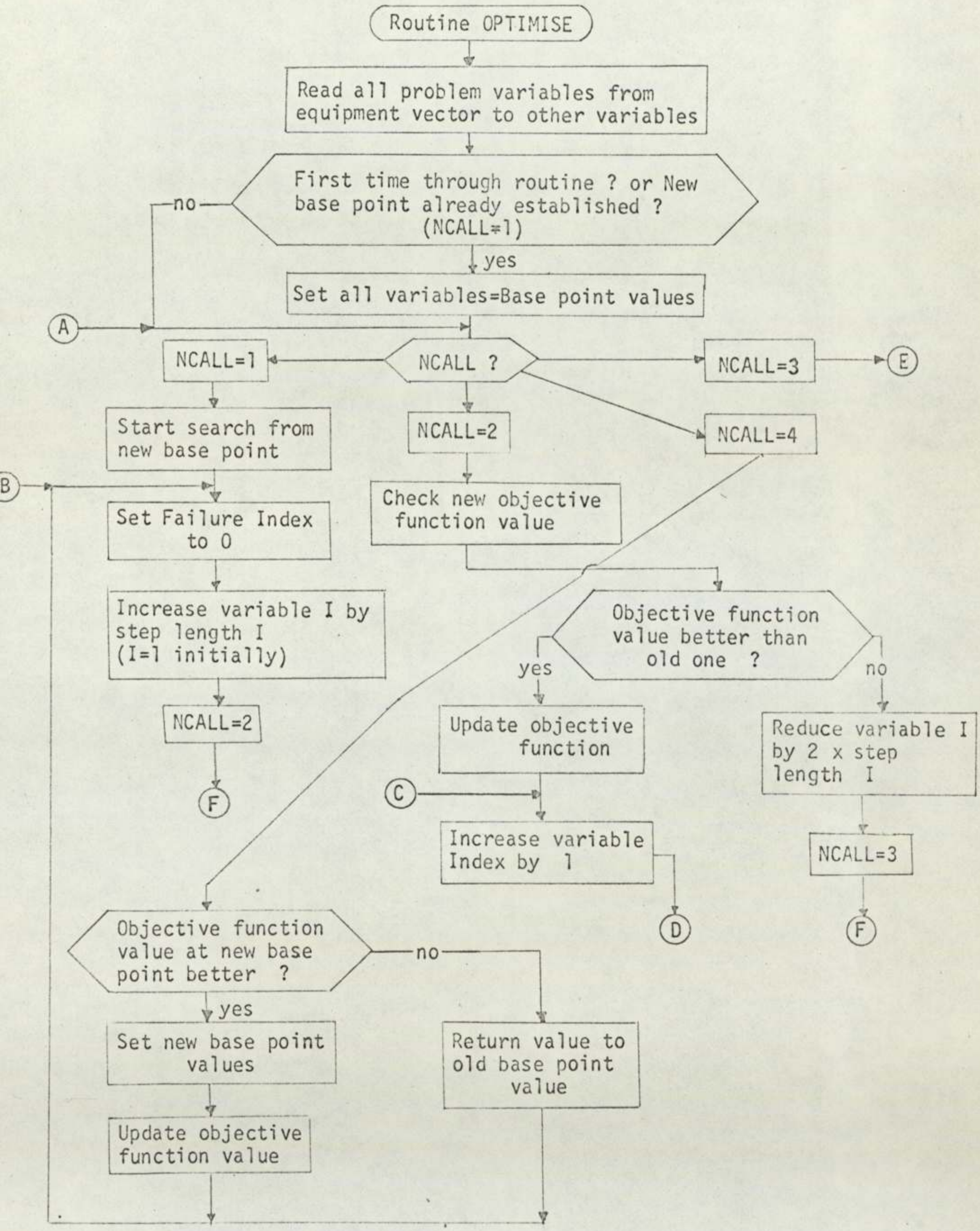
The number of variables - The number of inequality constraints - The number of equality constraints - The maximim number of steps - The initial step number - The starting variable - The starting unit number - The initial three variables values (unused variables are zeroed) - The equipment number containing the first design variable. The variable position in the vector - Same for the second and third variables - The three step lengths - The three minimum step lengths - Eleven zeroes - 0 or (+) number if a maximum objective function is analysed , (-) number for a minimum -

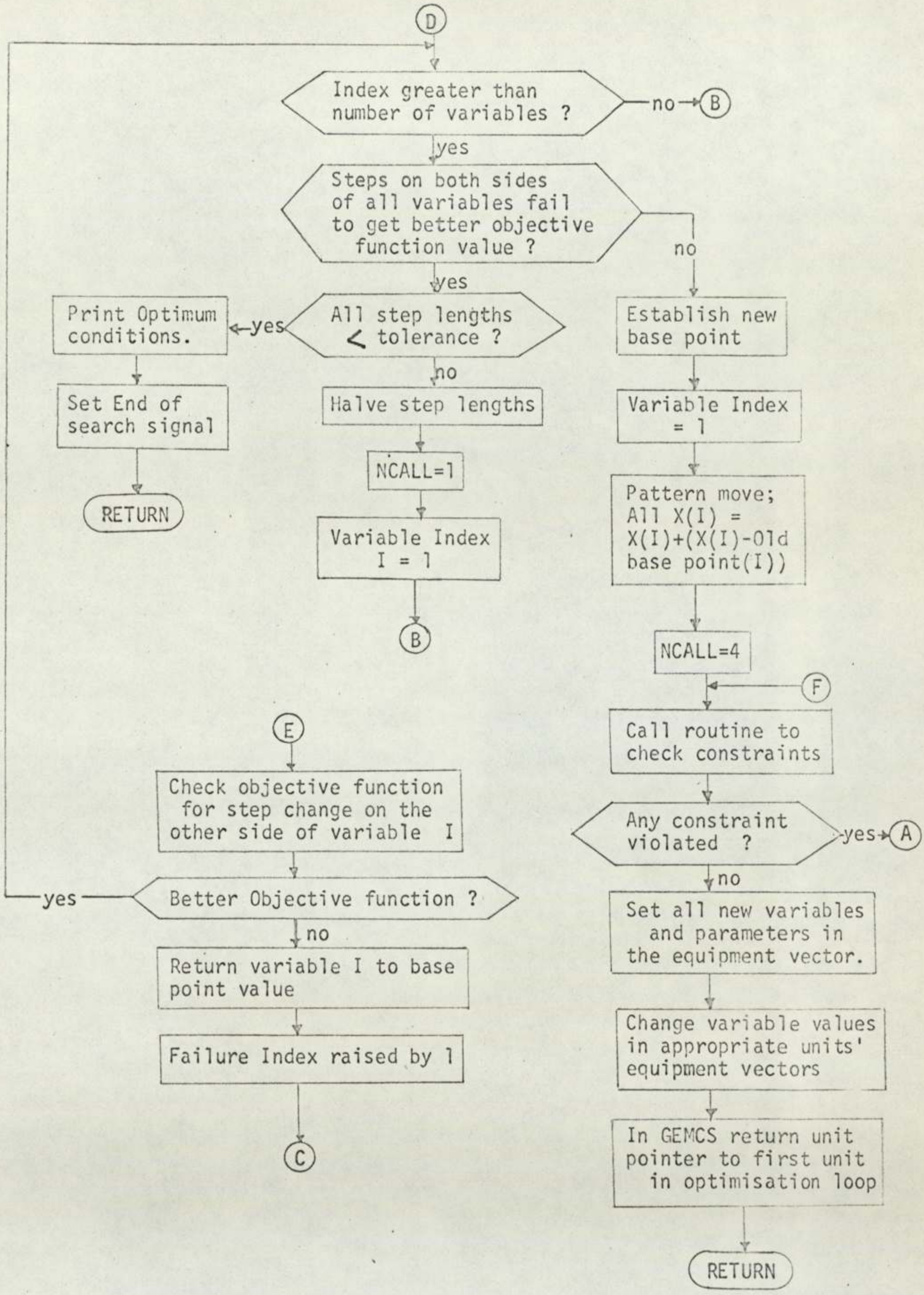
Number of units in the optimised sequence .

(Example of data input to GEMCS's version (example 2,chapter 3) :

3.	3.	0.	100.	1.
1.	1.	.08	.05	.02
1.	16.	2.	16.	3.
16.	.02	.02	.02	.001
.001	.001	0.	0.	0.
0.	0.	0.	0.	0.
0.	0.	0.	0.	5.)

Flowchart of routine OPTIMISE





Flowchart of routine OPTIMISE (Cont'd)

```
      DIMENSION X(3),XO(3),EQPT(3),VAR(3),STEPX(3),TOLX(3),XB(3)
      DIMENSION ENN(80),IFAIL(3)
      COMMON LLST(50),NS(100),EN(100),STRMI(4,30),STRMO(4,30),SN(17,30)
      COMMON IS,NE,JJ,LOOP,NIN,NOUT,MSN,ISP,NC,III
      COMMON KPRNT(10),NCALC,NOCOMP,NSR
      COMMON EEN(600),NPOINT(25,2),NCOUNT
      NVAR=EN(16)
      NCONS=EN(17)
      NEQUS=EN(18)
      MAXM=EN(19)
      M1=EN(20)
      IVAR=EN(21)
      NCALL=EN(22)
      XO(1)=EN(23)
      XO(2)=EN(24)
      XO(3)=EN(25)
      EQPT(1)=EN(26)
      EQPT(2)=EN(28)
      EQPT(3)=EN(30)
      VAR(1)=EN(27)
      VAR(2)=EN(29)
      VAR(3)=EN(31)
      NVIOL=EN(38)
      NFAIL=EN(39)
      UARTO=EN(41)
      X(1)=EN(42)
      X(2)=EN(43)
      X(3)=EN(44)
      XB(1)=EN(45)
      XB(2)=EN(46)
       XB(3)=EN(47)
      STEPX(1)=EN(32)
      STEPX(2)=EN(33)
      STEPX(3)=EN(34)
      TOLX(1)=EN(35)
      TOLX(2)=EN(36)
      TOLX(3)=EN(37)
      NFAYL=EN(48)
      TYPE=EN(49)
      NRET=EN(50)
      UART=STRMI(1,3)
```

```
      WRITE(2,600)UART
600   FORMAT(10X,5H UART,F20.///)
C
C     THIS IS THE DIRECT SEARCH ALGORITHM OF HOOK AND JEEVES
C
      NVOL1=1
      IF(NCALL-1)120,110,120
110   UARTO=UART
      DO 111 I=1,3
111   X(I)=XO(I)
120   CONTINUE
      IF(NVIOL)112,113,112
113   NVOL1=0
C     RETURN TO CORRECT SECTION OF ROUTINE
112   GO TO (170,200,210,355),NCALL
170   CONTINUE
C     MAKE SEARCH
180   NFAYL=0
190   CONTINUE
      X(IVAR)=X(IVAR)+STEPX(IVAR)
      NCALL=2
      GO TO 100
200   CONTINUE
      IF(TYPE)202,201,201
201   IF(UART-UARTO)205,205,230
202   IF(UARTO-UART)205,230,230
205   X(IVAR)=X(IVAR)-2.0*STEPX(IVAR)
      NCALL=3
      GO TO 100
210   CONTINUE
      IF(TYPE)212,211,211
211   IF(UART-UARTO)215,215,230
212   IF(UARTO-UART)215,215,230
215   NFAYL=NFAYL+1
      X(IVAR)=X(IVAR)+STEPX(IVAR)
      GO TO 240
230   UARTO=UART
240   IVAR=IVAR+1
      IF(IVAR-NVAR)190,190,250
250   IF(NFAYL-NVAR)315,260,315
260   DO 280 J=1,NVAR
```

```
      GO TO 385
290   DO 310 J=1,NVAR
310   STEPX(J)=STEPX(J)/2.0
      IVAR=1
      GO TO 180
C     ESTABLISH NEW BASE POINT
315   DO 320 I=1,NVAR
320   XB(I)=X(I)
      M1=M1+1
      IVAR=1
      IF(M1-MAXM)345,345,385
C     MAKE A PATTERN MOVE
345   DO 350 I=1,NVAR
350   X(I)=X(I)+(X(I)-XO(I))
      NCALL=4
      GO TO 100
355   CONTINUE
      IF(TYPE)357,356,356
356   IF(UART-UARTO)365,365,370
357   IF(UARTO-UART)365,365,370
365   DO 360 I=1,NVAR
      XO(I)=XB(I)
360   X(I)=XB(I)
      GO TO 180
370   DO 380 I=1,NVAR
380   XO(I)=XB(I)
      UARTO=UART
      GO TO 180
385   WRITE(2,3)(X(I),I=1,NVAR),UARTO
3     FORMAT(///10X,16H OPTIMUM REACHED //10X,31H INDEPENDANT VARIABLES
     1VALUES =  ,3F20.7,//10X,17H FUNCTION VALUE = ,F20.7///)
      LOOP=999
      RETURN
100   CONTINUE
      WRITE(2,601)(X(I),I=1,NVAR)
601   FORMAT(10X,5H X(I),3F20.7)
      CALL OPTI1(X,UART,NCONS,NEQUS,NVIOL,NFAIL,IFAIL)
      IF(UART)420,410,420
410   FN(20)=M1
      FN(21)=IVAR
      FN(22)=NCALL
```

```
      EN(23)=XO(1)
      EN(24)=XO(2)
      EN(25)=XO(3)
      EN(32)=STEPX(1)
      EN(33)=STEPX(2)
      EN(34)=STEPX(3)
      EN(38)=NVIOL
      EN(39)=NFAIL
      EN(40)=UART
      EN(41)=UARTO
      EN(42)=X(1)
      EN(43)=X(2)
      EN(44)=X(3)
      EN(45)=XB(1)
      EN(46)=XB(2)
      EN(47)=XB(3)
      EN(48)=NFAYL
      MM=EN(1)
      CALL DISKIO(2,MM)
      DO 500 I=1,60
      ENN(I)=EN(I)
500   EN(I)=0.
      DO 501  I=1,3
      J=24+2*I
      NIEN=IFIX(ENN(J))
      IF(NIEN)501,501,506
506   CALL DISKIO(1,NIEN)
      NEN=IFIX(ENN(J+1))
      EN(NEN)=X(I)
      CALL DISKIO(2,NIEN)
501   CONTINUE
      DO 502 I=1,60
      EN(I)=ENN(I)
502   ENN(I)=0.
      NC=NC-NRET
      RETURN
420   IF(TYPE)444,440,440
440   UART=-1.0E+20
      GO TO 112
444   UART=1.0E+20
      GO TO 112
```

```
      SUBROUTINE STAGES
      COMMON LLST(50),NS(100),EN(100),STRMI(4,30),STRMO(4,30),SN(17,30)
      COMMON IS,NE,JJ,LOOP,NIN,NOUT,MSN,ISP,NC,III
      COMMON KPRNT(10),NCALC,NOCOMP,NSR
      COMMON EEN(600),NPOINT(25,2),NCOUNT
      XI=STRMI(1,4)
      XO=EN(16)
      YI=STRMI(2,4)
      Q=STRMI(1,3)
      CALL YOUT(XO,YO)
      W=Q*(XI-XO)/(YO-YI)
      FUNC=Q*(XI-XO)*1.0-0.05*W
      STRMO(1,3)=Q
      STRMO(1,4)=XO
      STRMO(2,3)=W
      STRMO(2,4)=YO
      STRMO(3,3)=FUNC
      RETURN
      END
      SUBROUTINE YOUT(X,Y)
      IF(X-0.06)1,2,2
1     Y=2.42*X
      RETURN
2     IF(X-0.1)3,3,4
3     Y=0.1015+0.725*X
      RETURN
4     IF(X-0.185)5,6,6
5     Y=0.161+0.13*X
      RETURN
6     Y=X
      RETURN
      END
      SUBROUTINE OPTI1(X,UART,NCONS,NEQUS,NVIOL,NFAIL,IFAIL)
C
C     SEEK1 PENALTY FUNCTION
C
      DIMENSION X(3),PHI(3),PSI(3)
      DIMENSION IFAIL(3)
      NVIOL=0
      NFAIL=0
      IF(NCONS)12,12,4
```

```
4     DO 16 I=1,NCONS
16    PHI(I)=0.
      CALL CONST(X,NCONS,PHI)
      DO 1 I=1,NCONS
      IF(PHI(I))1,1,5
5     NVIOL=NVIOL+1
      IFAIL(I)=I
      NFAIL=NFAIL+1
1     CONTINUE
      IF(NVIOL)12,12,15
12    IF(NEQUS)15,115,7
7     DO 17 I=1,NEQUS
17    PSI(I)=0.0
      CALL EQUAL(X,PSI,NEQUS)
      DO 3 I=1,NEQUS
      IF(PSI(I))30,3,30
30    NVIOL=NVIOL+1
3     CONTINUE
115   IF(NVIOL)8,8,15
8     UART=0.0
      RETURN
15    UART=-1.0E+20
      RETURN
      END
      SUBROUTINE CONST(X,NCONS,PHI)
      DIMENSION X(3),PHI(3)
      SPLIT1=1.0-X(1)
      SPLIT2=1.0-X(2)
      SPLIT3=1.0-X(3)
      IF(SPLIT1)2,2,1
1     GO TO 3
2     PHI(1)=100.
3     IF(SPLIT2)4,4,5
4     PHI(2)=100.
5     IF(SPLIT3)6,6,7
6     PHI(3)=100.
7     RETURN
      END
      SUBROUTINE EQUAL(X,PSI,NEQUS)
      DIMENSION X(3),PSI(3)
      RETURN
```

APPENDIX THREE .

Contents :

Commands to compile and save the programs.

Listings of the programs making up the PEETPACK and PROPACK packages :

INTERACTER	
BATCHER	
PEETPACKSUBS	(Called PEETPACKS in source form)
EXECUTIVESUB	(Called EXECUTIVE in source form)
THERMOSUBS	(Called SUBS in source form)
MODELSUBS	(Called MODELS in source form)
PHYSUBS	(Called PHYSSUB in source form)
ECHOER	
LIBPROP	
PEETREADER	
PEETUPDATER	
BATCHREAD	
BATCHUPDATE	
INDPEETREAD	
INDPEETUP	
PROPACK	
PROPTS	
PPTEXTRA	

Commands to compile and save the various programs making up the PEETPACK and PROPACK packages, on the ICL-1905E computer at the University of Aston in Birmingham :

1 - <u>INTERACTER</u> :

```
CE INTERACTBIN(*DA,BUCK 1,KWØR 10)
UAFØRTRAN PRØG INTERACTER,ØWNPD,LINES 2000,-
DUMPØN INTERACTBIN,NØRUN,EXIT
```

2 - <u>BATCHER</u> :

```
UAFØRTRAN PRØG BATCHER,ØWNPD,LINES 2000,-
SAVE BATCHBIN,NØRUN,EXIT
```

3 - <u>PEETPACK</u> :

```
UAFØRTRAN PRØG PEETPACKS,ØWNPD,LINES 3000,-
LIB :ECP0750.EXECUTIVESUB,LIB :ECP0750.MØDELSUBS,-
LIB :ECP0750.PHYSUBS,LIB :ECP0750.THERMØSUBS,-
SAVE PEETPACKBIN,NØRUN,EXIT
```

4 - <u>ECHØER</u> :

```
UAFØRTRAN PRØG ECHØER,ØWNPD,LINES 2000,-
SAVE ECHØERBIN,NØRUN,EXIT
```

5 - <u>PRØPACK</u> :

```
UAFØRTRAN PRØG LIBPRØP,ØWNPD,LINES 2000,-
LIB :ECP0750.PHYSUBS,LIB :ECP0750.THERMØSUBS,-
SAVE PRØPBIN,NØRUN,EXIT
```

Program INTERACTER

```
      OVERLAY PROGRAM(NEWINT)
      OVERLAY(1,1)INTERACT
      OVERLAY(1,2)ORDER,FLAGER,EQCALLER
      INPUT 1 = CR0
      INPUT 3 = CR1
      INPUT 5 = CR2
      INPUT 6 = CR3
      INPUT 9 = CR4
      OUTPUT 2 = LP0
      OUTPUT 4 = CP0
      USE 7 = /ARRAY
      USE 8 = /ARRAY
      TRACE 2
      END
C
      MASTER
C
      NWF=4
      IRETURN=0
10    CALL INTERACT(IRETURN,NR,NW)
      IF(IRETURN.LT.1)GO TO 20
      CALL ORDER(NR,NW)
      IRETURN=1
      GO TO 10
20    IF(IRETURN.EQ.(-1))CALL EQCALLER(NWF)
      STOP
      END
C
      SUBROUTINE INTERACT(IRETURN,NR,NW)
C
      DIMENSION
     *TITLE(10),BUFFER(10),VNAMES(35),A(20),B(20),
     *EEN(500),NEMAT(30,2),SMATRX(400),RNAME(30),VNAME(30),MODULE(30),
     *PROPRT(10,25),PRTEXTRA(10,35),USERPPT(10,10),SMODN(20),FORMS(2)
     *,VNAME2(6)
     *,ERR(30)
C
      COMMON
     *KPM(30,8),KSFLAG(30),KSEM(30,3),NELIST(30,2),KPS(10),
     *NEMAX,NSMAX,NCOLPM,NLIST,NPS,LOOPN,ENAME(30),SNAME(30)
```

```
C
      DATA VNAMES/
     *8HYES     ,8HNO      ,4HLOOP,3HSEQ,
     18HTYPE    ,8HCHANGE  ,8HEXIT    ,8HEND     ,8HCALCULAT,
     28HDATA    ,8HPRINT   ,8HINPUT   ,8HADD     ,8HTRACE   ,
     38HACCELERA,8HTOLERANC,8HLOOPS   ,8HBRITISH ,8HS.I.    ,
     48HKEEPON  ,8HNOGO    ,8HALL     ,8HORDER   ,8HUNIT    ,
     58HSTREAM  ,8HROUTINE ,8HPREFERRE,8HPROPERTI,8HCONTROLS,
     68HCOMPONEN,8HPARAMETE,8HROUTINES,8HSTREAMS ,8H        ,
     78HERRORS
     */
      DATA VNAME(1)/
     *80HFLOW    (MOLAR)  TEMPERATURE     PRESSURE  ABS.  ENTHALPY
     * VAPOUR FRACTION
     */
C
      DATA VNAME2(1)/
     *48HURE DEGF  PSIA  BTU/LBMLURE DEGC  BARS   KJ/KGML  /
C
      DATA SMODN /
     *8HMASSMIX ,8HMIXER   ,8HSPLITTER,8HSPLITMIX,8HFLASHER ,
     *8HPURIFYER,8HSETBP   ,8HSETDP   ,8HHEATER  ,8HHEATEXCH,
     *8HPUMP    ,8HCOMPRESR,8HVALVE   ,8HDISCOL  ,8HREACTOR ,
     *8HUNIT1   ,8HUNIT2   ,8HUNIT3   ,8HUNIT4   ,8HUNIT5
     */
C
      DATA ZERO/8H00000000/
C
C
C
C
C ***   INTERACTIVE  PROCESS  DATA  READER
C
C
C *** INPUT/OUTPUT CHANNELS
C
      NR=1
      NW=2
      NRF=5
      NRCOMP=3
      NRCOMP1=6
      NRCOMPX=9
```

```
      NWF=4
      IRM=0
      IRT=0
C
C
      NCOMND=18
C
      IF(IRETURN.EQ.1)GO TO 3530
      WRITE(NW,4500)
C
C *** BUFFER AREA AND SYSTEM CHECKS
C
      CALL DEFBUF(7,80,BUFFER)
      CALL DEFBUF(8,16,FORMS)
      I=1
      CALL SSWTCH(I,J)
      I=2
      CALL SSWTCH(I,JSWTCH2)
      IF(J.EQ.1)GO TO 50
C
C *** INITIAL VALUES AND FLAGS
C
      NEMAX=0
      NSMAX=0
      NOCOMP=0
      NUSRPPT=0
      NCMORE=0
      NCTOT=0
      NPAR=0
      NCOLPM=0
      NPS=0
      NSLMAX=5
      NVNAME=10
      NSLMX1=6
C
      KONVRG=0
      ITRACE=-1
      IUNITS=0
      NOGO=1
      LOOPN=10
```

```
C
      INDEX=0
      NLIST=0
      NIE=0
      NNE=1
      ISEQ=1
      ISQ=0
      INW=0
      NE=0
      KALL=0
      IRT=0
      IERR=0
C
C
      WRITE(NW,4510)
      READ(NR,4000)TITLE
      GO TO 200
C
C
C *** INTERACTIVE UPDATER
C
C
C *** READ STORED DATA AND RESULTS FROM LOG FILE
C
50    READ(NRF,4000)TITLE
      READ(NRF,4010)NEMAX,NSMAX,NOCOMP,NSLMAX,NPAR,NCOLPM,KONVRG,
     *NPS,ITRACE,NLOOPS,NLIST,NVNAME,NPROP,NOGO,IUNITS,NUSRPPT,NCTOT
     *,NPROPX,IRT,IERR
      READ(NRF,4020)TOL
      IF(NEMAX.EQ.0)GO TO 54
      READ(NRF,4000)(SNAME(I),I=1,NSMAX)
      READ(NRF,4000)(ENAME(I),I=1,NEMAX)
      IF(IRT.EQ.0)GO TO 52
      READ(NRF,4000)(RNAME(I),I=1,NEMAX)
      READ(NRF,4010)(MODULE(I),I=1,NEMAX)
52    READ(NRF,4000)(VNAME(I),I=1,NVNAME)
54    IF(NOCOMP.EQ.0)GO TO 58
      DO 56 I=1,NOCOMP
      READ(NRF,4000)PROPRT(I,1),PROPRT(I,2)
      READ(NRF,4020)(PROPRT(I,J),J=3,NPROP)
56    READ(NRF,4020)(PRTEXTRA(I,J),J=1,NPROPX)
58    IF(NUSRPPT.EQ.0)GO TO 62
```

```
62    IF(NEMAX.EQ.0)GO TO 64
      READ(NRF,4010)((KPM(I,J),J=1,NCOLPM),I=1,NEMAX)
      READ(NRF,4010)((KSEM(I,J),J=1,5),I=1,NSMAX)
64    IF(NSMAX.EQ.0)GO TO 68
      READ(NRF,4020)(SMATRX(I),I=1,NSLMAX*NSMAX)
      IF(NPAR.EQ.0)GO TO 68
      READ(NRF,4020)(EFN(I),I=1,NPAR)
      READ(NRF,4010)((NEMAT(I,J),J=1,2),I=1,NEMAX)
68    IF(NLIST.EQ.0)GO TO 70
      READ(NRF,4010)((NELIST(I,J),J=1,2),I=1,NLIST)
      READ(NRF,4010)(KSFLAG(I),I=1,NSMAX)
70    IF(NPS.EQ.0)GO TO 80
      READ(NRF,4010)(KPS(I),I=1,NPS)
80    IF(IERR.EQ.0)GO TO 110
      READ(NRF,4020)(ERR(I),I=1,NSMAX)
C
C
C *** INTERACTIVE UPDATING COMMANDS
C
110   KALL=0
      WRITE(NW,115)
115   FORMAT(/2X,10H COMMAND ?  )
      READ(NR,4000)BUFFER
      N=-1
      CALL COPVER(8,A,BUFFER,N)
      DO 120 I=1,NCOMND
      CALL COMP8(A(1),VNAMES(I+4),ICOMP)
      IF(ICOMP.NE.1)GO TO 120
      INDEX=I
      GO TO 140
120   CONTINUE
130   GO TO 1035
140   GO TO (
     * 1500,1200,3000, 110,2700
     *,2000,1800,1200,1200, 900
     *, 905, 915, 920, 925, 930
     *, 935, 940
     *),INDEX
C
C *** PROCESS FLOWSHEET
```

```
C
200   WRITE(NW,205)
205   FORMAT(/4X,23H PROCESS FLOWSHEET DATA //2X,
     *31H NUMBER OF UNITS IN THE PLANT :)
      I=0
      READ(NR,4040)AN
      NEMAX=AN
208   I=I+1
      NCOLMX=1
      WRITE(NW,210)
210   FORMAT(/2X,12H UNIT NAME :  )
      READ(NR,4000)BUFFER
      N=1
      CALL COPYER(8,A,BUFFER,N)
      IF(NE.EQ.0)GO TO 230
212   DO 215 J=1,NE
      CALL COMP8(A(1),ENAME(J),ICOMP)
      IF(ICOMP.EQ.1)GO TO 220
215   CONTINUE
      GO TO 230
220   WRITE(NW,225)
225   FORMAT(/2X,15H MODIFICATION !  /)
226   NU=J
      I=I-1
      GO TO 235
230   NE=NE+1
      IF(KALL.LT.10)GO TO 234
      NEMAX=NEMAX+1
      NE=NEMAX
234   NU=NE
      CALL COPY8(ENAME(NE),A(1))
235   DO 236 JK=1,NCOLPM
236   KPM(NU,JK)=0
      WRITE(NW,240)
      IPM=1
240   FORMAT(2X,16H INPUT STREAMS :)
      READ(NR,4000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
      IF(N.EQ.0)GO TO 265
      DO 260 K=1,N
      IF(NSMAX.EQ.0)GO TO 250
```

```
      IF(ICOMP.EQ.1)GO TO 255
  245 CONTINUE
  250 NSMAX=NSMAX+1
      CALL COPY8(SNAME(NSMAX),A(K))
      J=NSMAX
  255 NCOLMX=NCOLMX+1
      KPM(NU,NCOLMX)=J
  260 CONTINUE
  265 KPM(NU,1)=NU
      WRITE(NW,270)
  270 FORMAT(2X,17H OUTPUT STREAMS : )
      READ(NR,4000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
      IF(N.EQ.0) GO TO 295
      DO 290 K=1,N
      IF(NSMAX.EQ.0)GO TO 280
      DO 275 J=1,NSMAX
      CALL COMP8(SNAME(J),A(K),ICOMP)
      IF(ICOMP.EQ.1)GO TO 285
  275 CONTINUE
  280 NSMAX=NSMAX+1
      CALL COPY8(SNAME(NSMAX),A(K))
      J=NSMAX
  285 NCOLMX=NCOLMX+1
      KPM(NU,NCOLMX)=-J
  290 CONTINUE
  295 IF(NCOLMX.GT.NCOLPM)NCOLPM=NCOLMX
      NLIST=0
      IF(KALL.EQ.10)GO TO 110
      IF(I.LT.NEMAX)GO TO 208
      IF(KALL.EQ.11)GO TO 110
      ASSIGN 3300 TO KEXIT
      GO TO 1020
C
C *** EQUIPMENT PARAMETERS
C
  300 WRITE(NW,305)
  305 FORMAT(/4X,19H UNITS PARAMETERS :/)
      I=1
```

```
307    WRITE(NW,310)ENAME(I)
310    FORMAT(/2X,9H UNIT :  ,A8/2X,22H HOW MANY PARAMETERS ?)
       READ(NR,4040)AN
       J=AN
       IF(J.LT.15)GO TO 325
       WRITE(NW,315)
315    FORMAT(4X,15H ARE YOU SURE ?)
       READ(NR,4000)BUFFER
       ASSIGN 320 TO KALLER
       GO TO 1000
320    IF(NO.EQ.1)GO TO 307
325    NEMAT(I,2)=J
       IF(J.EQ.0)GO TO 340
       NEMAT(I,1)=NPAR+1
       NPAR=NPAR+J
       WRITE(NW,330)
330    FORMAT(2X,13H PARAMETERS :  )
       READ(NR,4060)(EEN(K),K=NEMAT(I,1),NPAR)
       IF(KALL.EQ.10)GO TO 110
340    I=I+1
       IF(I.LE.NEMAX)GO TO 307
       IF(KALL.EQ.11)GO TO 110
       IF(KALL.EQ.12)GO TO 2730
       ASSIGN 400 TO KEXIT
       GO TO 1020
C
C *** ROUTINE NAMES
C
400    I=0
401    WRITE(NW,405)
405    FORMAT(/4X,20H UNITS MODEL NAMES :     )
406    WRITE(NW,415)
415    FORMAT(/2X,42H TYPE THE MODEL NAME AFTER THE UNIT NAME :/)
418    I=I+1
       WRITE(NW,445)ENAME(I)
420    READ(NR,4000)BUFFER
       N=1
       CALL COPYER(8,A,BUFFER,N)
       IPM=1
       IF(JSWTCH2.EQ.1)GO TO 440
       DO 425 J=1,21
       CALL COMP8(A(1),SMODN(J),ICOMP)
```

```
      WRITE(NW,430)
430   FORMAT(/2X,58H MODEL REQUESTED NOT IN LIBRARY, PLEASE CHOOSE ANOTH
     *ER ONE  /)
      GO TO 420
435   MODULE(I)=J
440   CALL COPY8(RNAME(I),A(1))
445   FORMAT(2X,A8)
      IF(KALL.EQ.10)GO TO 110
      IF(I.LT.NEMAX)GO TO 418
      IRT=1
      IF(KALL.EQ.11)GO TO 110
      IF(KALL.EQ.12)GO TO 2740
      ASSIGN 500 TO KEXIT
      GO TO 1020
C
C *** COMPONENTS
C
500   NC=1
      WRITE(NW,505)
505   FORMAT(/4X,13H COMPONENTS : /)
      WRITE(NW,510)
510   FORMAT(2X,37H TOTAL NUMBER OF LIBRARY COMPONENTS :)
      READ(NR,4040)AN
      NOCOMP=AN
      NCTOT=NOCOMP
      IF(NOCOMP.EQ.0)GO TO 575
      K=9
      WRITE(NW,515)
515   FORMAT(/2X,27H LIBRARY COMPONENTS NAMES :/)
520   READ(NR,4000)BUFFER
      N=-1
      CALL COPYER(16,A,BUFFER,N)
      J=-1
      N=N/2
      NC1=NC+N-1
      DO 525 I=NC,NC1
      K=K+2
      J=J+2
      CALL COPY8(VNAME(K),A(J))
      CALL COPY8(VNAME(K+1),A(J+1))
```

```
      CALL COPY8(PROPRT(I,1),A(J))
525   CALL COPY8(PROPRT(I,2),A(J+1))
      NC=NC+N
      IF(NC.LE.NOCOMP)GO TO 560
      NC=0
530   PAUSE PROPT
      GO TO 535
535   READ(NRCOMP,4010)NCOMP
      READ(NRCOMP,4010)NPROP
      READ(NRCOMPX,4010)NCOMPX
      READ(NRCOMPX,4010)NPROPX
      DO 560 I=1,NCOMP
      READ(NRCOMP,4000)A(1),A(2)
      READ(NRCOMPX,4000)A(3)
      DO 540 J=1,NOCOMP
      CALL COMP8(A(1),PROPRT(J,1),ICOMP)
      IF(ICOMP.NE.1)GO TO 540
      CALL COMP8(A(2),PROPRT(J,2),ICOMP)
      IF(ICOMP.EQ.1)GO TO 545
540   CONTINUE
      GO TO 550
545   IF(PROPRT(J,3).GT.0.0)GO TO 550
      NC=NC+1
      READ(NRCOMP,4050)(PROPRT(J,M),M=3,NPROP)
      READ(NRCOMPX,4050)(PRTEXTRA(J,M),M=1,NPROPX)
      GO TO 559
550   DO 555 M=3,NPROP
555   READ(NRCOMP,4050)Z
      DO 556 M=1,NPROPX
556   READ(NRCOMPX,4050)ZZ
559   IF(NC.EQ.NOCOMP)GO TO 575
560   CONTINUE
      IF(NC.GE.NOCOMP)GO TO 575
      DO 570 I=1,NOCOMP
      IF(PROPRT(I,3).GT.0.0)GO TO 570
      WRITE(NW,565)PROPRT(I,1),PROPRT(I,2)
565   FORMAT(/2X,2A8,2X,43H IS NOT A RECOGNISED LIBRARY COMPONENT NAME/
     *2X,17H PLEASE CHANGE IT/)
      READ(NR,4000)BUFFER
      N=2
      CALL COPYER(16,A,BUFFER,N)
      CALL COPY8(PROPRT(I,1),A(1))
```

```
      CALL COPY8(VNAME(J),A(1))
      CALL COPY8(VNAME(J+1),A(2))
570   CONTINUE
      GO TO 530
575   NSLMAX=NSLMAX+NOCOMP
577   NVNAME=NVNAME+NOCOMP+NOCOMP
      GO TO 800
C
C *** TYPE OF UNITS USED (BRITISH OR S.I.)
C
580   WRITE(NW,585)
585   FORMAT(/2X,40H WHICH TYPE OF UNITS ? (BRITISH OR S.I.))
590   READ(NR,4000)BUFFER
      N=1
      CALL COPYER(8,A,BUFFER,N)
      CALL COMP8(A(1),VNAMES(18),ICOMP)
      IF(ICOMP.EQ.1)GO TO 596
      CALL COMP8(A(1),VNAMES(19),ICOMP)
      IF(ICOMP.EQ.1)GO TO 597
      WRITE(NW,595)
595   FORMAT(/2X,35H UNRECOGNISED ANSWER, PLEASE RETYPE/)
      GO TO 590
596   IUNITS=0
      GO TO 598
597   IUNITS=1
598   I3=IUNITS*3
      CALL COPY8(VNAME(4),VNAME2(I3+1))
      CALL COPY8(VNAME(6),VNAME2(I3+2))
      CALL COPY8(VNAME(8),VNAME2(I3+3))
      IF(IUNITS.EQ.0)GO TO 5990
      DO 599 I=1,NOCOMP
      PROPRT(I,5)=PROPRT(I,5)/14.5
      PROPRT(I,6)=PROPRT(I,6)/1.8
      PROPRT(I,7)=PROPRT(I,7)/16.03
      PROPRT(I,8)=PROPRT(I,8)/1.8
      PRTEXTRA(I,1)=PRTEXTRA(I,1)/1.8
      IF(PRTEXTRA(I,1).EQ.0.0)PRTEXTRA(I,3)=(PRTEXTRA(I,3)-32.0)/1.8
      PRTEXTRA(I,8)=PRTEXTRA(I,8)/1.8
      PRTEXTRA(I,29)=PRTEXTRA(I,29)/1.8
      DO 599 J=14,19
```

```
      PROPRT(I,J)=PROPRT(I,J)*2.325
590   CONTINUE
5990  IF(KALL.EQ.12)GO TO 2750
      IF(KALL.GT.9)GO TO 110
      ASSIGN 600 TO KEXIT
      GO TO 1020
C
C *** STREAMS SPECIFICATIONS
C
600   I=0
601   WRITE(NW,605)
605   FORMAT(/4X,25H STREAMS SPECIFICATIONS : /
     *2X,16H INPUT STREAMS :)
607   I=I+1
      IF(KSEM(I,2).GT.0)GO TO 625
      WRITE(NW,610)SNAME(I)
610   FORMAT(/2X,8H STREAM  ,A8/)
612   K=-1
      I1=(I-1)*NSLMAX
      DO 620 J=1,NSLMAX
      K=K+2
      WRITE(NW,615)VNAME(K),VNAME(K+1)
615   FORMAT(2X,2A8,3H  :)
620   READ(NR,4040)SMATRX(I1+J)
      IF(KALL.GT.9.AND.KALL.LT.12)GO TO 110
625   IF(I.LT.NSMAX)GO TO 607
      WRITE(NW,635)
635   FORMAT(/2X,18H ITERATE STREAMS : /)
      DO 645 I=1,NSMAX
      IF(KSFLAG(I).NE.2)GO TO 645
      WRITE(NW,610)SNAME(I)
      K=-1
      I1=(I-1)*NSLMAX
      DO 640 J=1,NSLMAX
      K=K+2
      WRITE(NW,615)VNAME(K),VNAME(K+1)
640   READ(NR,4040)SMATRX(I1+J)
645   CONTINUE
      IF(KALL.GT.9)GO TO 110
      ASSIGN 700 TO KEXIT
      GO TO 1020
C
```

```
700   IF(LOOPN.LT.1)GO TO 735
      WRITE(NW,705)
705   FORMAT(/4X,22H SIMULATION CONTROLS :/)
      WRITE(NW,710)
710   FORMAT(2X,31H MAXIMUM PERCENTAGE TOLERANCE :)
      READ(NR,4040)TOL
      TOL=TOL/100.
      WRITE(NW,715)
715   FORMAT(2X,26H MAXIMUM NUMBER OF LOOPS : )
      READ(NR,4040)AN
      NLOOPS=AN
      WRITE(NW,720)
720   FORMAT(2X,37H IS A CONVERGENCE PROMOTER REQUIRED ?)
      ASSIGN 725 TO KALLER
      READ(NR,4000)BUFFER
      GO TO 1000
725   IF(NO.EQ.1)GO TO 735
      WRITE(NW,730)
730   FORMAT(2X,19H TYPE OF PROMOTER :)
      READ(NR,4040)AN
      KONVRG=AN
735   WRITE(NW,740)
740   FORMAT(2X,22H IS TRACING REQUIRED ?)
      READ(NR,4000)BUFFER
      ASSIGN 745 TO KALLER
      GO TO 1000
745   IF(NO.EQ.1)GO TO 755
      WRITE(NW,750)
750   FORMAT(2X,23H FREQUENCY OF TRACING :)
      READ(NR,4040)AN
      ITRACE=AN
755   IF(LOOPN.LT.1)GO TO 770
      WRITE(NW,760)
760   FORMAT(2X,26H IS NOGO OPTION REQUIRED ?)
      READ(NR,4000)BUFFER
      ASSIGN 765 TO KALLER
      GO TO 1000
765   NOGO=1-NO
      GO TO 780
770   WRITE(NW,775)
```

```
775   FORMAT(2X,31H PLANT FLOW DIAGRAM SEQUENTIAL /
     *2X,31H NO CONTROL PARAMETERS REQUIRED  /)
780   ASSIGN 110 TO KEXIT
      GO TO 1020
C
C *** USER-PROVIDED PROPERTIES
C
800   WRITE(NW,815)
815   FORMAT(/2X,66H DO YOU WISH TO ADD OTHER COMPONENTS NOT IN THE PEET
     *PACK LIBRARY ?)
      READ(NR,4000)BUFFER
      ASSIGN 820 TO KALLER
      GO TO 1000
820   IF(NO.EQ.1)GO TO 851
      WRITE(NW,825)
825   FORMAT(/2X,16H HOW MANY MORE ?)
      READ(NR,4040)AN
      NCMORE=AN
      WRITE(NW,830)
830   FORMAT(/2X,19H COMPONENTS NAMES :)
      L=0
835   READ(NR,4000)BUFFER
      N=-1
      CALL COPYER(16,A,BUFFER,N)
      J=NVNAME-1
      K=NOCOMP+L
      DO 840 I=1,N,2
      K=K+1
      J=J+2
      CALL COPY8(VNAME(J),A(I))
      CALL COPY8(VNAME(J+1),A(I+1))
      CALL COPY8(PROPRT(K,1),A(I))
840   CALL COPY8(PROPRT(K,2),A(I+1))
      NVNAME=NVNAME+N
      N=N/2
      L=L+N
      IF(L.LT.NCMORE)GO TO 835
      NC=0
842   PAUSE USRPT
      GO TO 843
843   READ(NRCOMP1,4010)NCOMP
      NPROP1=NPROP
```

```
      READ(NRCOMP1,4000)A(1),A(2)
      K=NOCOMP
      DO 844 J=1,NCMORE
      K=K+1
      CALL COMP8(A(1),PROPRT(K,1),ICOMP)
      IF(ICOMP.NE.1)GO TO 844
      CALL COMP8(A(2),PROPRT(K,2),ICOMP)
      IF(ICOMP.EQ.1)GO TO 845
844   CONTINUE
      GO TO 846
845   IF(ABS(PROPRT(K,3)).GT.0.0)GO TO 846
      NC=NC+1
      READ(NRCOMP1,4050)(PROPRT(K,M),M=3,NPROP)
      GO TO 848
846   DO 847 M=3,NPROP
847   READ(NRCOMP1,4050)ZZ
848   CONTINUE
      IF(NC.GE.NCMORE)GO TO 850
      DO 849 I=NOCOMP+1,NOCOMP+NCMORE
      IF(ABS(PROPRT(I,3)).GT.0.0)GO TO 849
      WRITE(NW,565)PROPRT(I,1),PROPRT(I,2)
      READ(NR,4000)BUFFER
      N=2
      CALL COPVER(8,A,BUFFER,N)
      CALL COPY8(PROPRT(I,1),A(1))
      CALL COPY8(PROPRT(I,2),A(2))
      J=(NSLMAX+I)*2-1
      CALL COPY8(VNAME(J),A(1))
      CALL COPY8(VNAME(J+1),A(2))
849   CONTINUE
      GO TO 842
850   NVNAME=NVNAME+NCMORE+NCMORE
      NCTOT=NOCOMP+NCMORE
      NSLMAX=NSLMAX+NCMORE
      IF(NPROP1.GT.NPROP)NPROP=NPROP1
851   WRITE(NW,852)
852   FORMAT(/2X,49H DO YOU WISH TO INPUT ANY ADDITIONAL PROPERTIES ?)
      READ(NR,4000)BUFFER
      ASSIGN 853 TO KALLER
      GO TO 1000
```

```
853   IF(NO.EQ.1)GO TO 890
      WRITE(NW,855)
855   FORMAT(/2X,54H HOW MANY OTHER PROPERTIES FOR COMPONENTS IN LIBRARY
     *? )
      READ(NR,4040)AN
      NUSRPPT=AN
      IF(NUSRPPT.EQ.0)GO TO 875
      WRITE(NW,860)
860   FORMAT(/2X,42H TYPE PROPERTY DATA AFTER COMPONENTS NAMES /)
      DO 870 I=1,NOCOMP
      WRITE(NW,865)PROPRT(I,1),PROPRT(I,2)
865   FORMAT(2X,2A8)
      READ(NR,4060)(USERPPT(I,J),J=1,NUSRPPT)
870   CONTINUE
875   IF(NCMORE.EQ.0)GO TO 890
      WRITE(NW,880)
880   FORMAT(/2X,47H HOW MANY OTHER PROPERTIES FOR NEW COMPONENTS ?)
      READ(NR,4040)AN
      NUSR2=AN
      IF(NUSR2.LT.1)GO TO 580
      WRITE(NW,860)
      L=NVNAME-2*NCMORE-1
      DO 885 I=1,NCMORE
      L=L+2
      J=NOCOMP+I
      WRITE(NW,865)VNAME(L),VNAME(L+1)
885   READ(NR,4060)(USERPPT(J,K),K=1,NUSR2)
      IF(NUSR2.GT.NUSRPPT)NUSRPPT=NUSR2
890   IS=1
      CALL SSWTCH(JS,IS)
      IF(IS.EQ.1)WRITE(NW,895)
895   FORMAT(/2X,43H MAKE SURE YOU UPDATE YOUR STREAM GUESSES !/)
      GO TO 580
C
C *** TRACE
C
900   IF(N.GT.1)GO TO 903
      ITRACE=0
      GO TO 110
903   N=-2
      CALL COPYER(8,A,BUFFER,N)
      ITRACE=A(2)
```

```
C *** CONVERGENCE PROMOTER
C
905   IF(N.GT.2)GO TO 910
      KONVRG=1
      GO TO 110
910   N=-3
      CALL COPYER(8,A,BUFFER,N)
      KONVRG=A(3)
      GO TO 110
C
C *** TOLERANCE
C
915   N=-3
      CALL COPYER(8,A,BUFFER,N)
      TOL=A(3)/100.
      GO TO 110
C
C *** NUMBER OF LOOPS
C
920   N=-2
      CALL COPYER(8,A,BUFFER,N)
      NLOOPS=A(2)
      GO TO 110
C
C *** TYPE OF UNITS USED
C
C     BRITISH UNITS
925   IUNITS=0
      KALL=10
      GO TO 598
C     S.I. IUNITS
930   IUNITS=1
      KALL=10
      GO TO 598
C
C *** OPTION TO GO ON OR STOP IF A LOOP FAILS TO CONVERGE IN THE
C *** SPECIFIED NUMBER OF LOOPS
C
C     GO ON (KEEPON) OPTION
935   NOGO=0
```

```
      GO TO 110
C     STOP RUN (NOGO) OPTION
940   NOGO=1
      GO TO 110
C
C *** CHECK ON YES/NO ANSWER
C
1000  N=1
      CALL COPYER(8,A,BUFFER,N)
      CALL COMP8(A(1),VNAMES(1),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1010
      CALL COMP8(A(1),VNAMES(2),ICOMP)
      IF(ICOMP.EQ.1) GO TO 1015
      WRITE(NW,1005)
1005  FORMAT(/2X,21H PLEASE RETYPE ANSWER/)
      READ(NR,4000)BUFFER
      GO TO 1000
C *** NO = 0  IF ANSWER IS  YES
1010  NO=0
      GO TO KALLER
C *** NO = 1  IF ANSWER IS  NO
1015  NO=1
      GO TO KALLER
C
C *** EXIT OPTION
C
1020  WRITE(NW,1025)
1025  FORMAT(/2X,7H EXIT ?)
      READ(NR,4000)BUFFER
      ASSIGN 1030 TO KALLER
      GO TO 1000
1030  IF(NO.EQ.0)GO TO 3000
      GO TO KEXIT
C
C *** ERROR MESSAGES
C
1035  WRITE(NW,1040)
1040  FORMAT(/2X,36H UNRECOGNISED COMMAND, PLEASE RETYPE/)
      GO TO 110
1045  WRITE(NW,1050)
1050  FORMAT(//2X,40H COMMAND NOT YET ALLOWED IN USER CONTEXT//)
      GO TO 110
```

```
C
1200  DO 1205 I=1,10
      CALL COMP8(A(2),VNAMES(21+I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1210
1205  CONTINUE
      GO TO 1035
1210  KALL=10
      GO TO (1400,3300,1300,1215,1235,3500, 800,700,1350,1250),I
C
C *** CHANGE STREAM ---
C
1215  DO 1220 I=1,NSMAX
      CALL COMP8(A(N),SNAME(I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1228
1220  CONTINUE
      J=1
1222  WRITE(NW,1225)A(N)
1225  FORMAT(/2X,A8,2X,63H IS NOT A RECOGNISED STREAM NAME, PLEASE RETYP
     PE THE STREAM NAME/)
      READ(NR,4000)BUFFER
      N=1
      CALL COPYER(8,A,BUFFER,N)
      GO TO (1215,1515),J
1228  WRITE(NW,1229)
1229  FORMAT(/)
      GO TO 612
C
C *** CHANGE ROUTINE ---
C
1235  DO 1240 I=1,NEMAX
      CALL COMP8(A(N),ENAME(I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1245
1240  CONTINUE
      ASSIGN 1235 TO KTYPE
1242  WRITE(NW,1244)A(N)
1244  FORMAT(/2X,A8,2X,60H IS NOT A RECOGNISED UNIT NAME, PLEASE RETYPE
     * THE UNIT NAME /)
      READ(NR,4000)BUFFER
      N=1
      CALL COPYER(8,A,BUFFER,N)
```

```
      GO TO KTYPE
1245  I=I-1
      GO TO 406
C
C *** CHANGE PARAMETERS ---
C
1250  CALL COMP8(A(1),VNAMES(12),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1310
      DO 1255 I=1,NEMAX
      CALL COMP8(A(N),ENAME(I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1260
1255  CONTINUE
      ASSIGN 1250 TO KTYPE
      GO TO 1242
1260  WRITE(NW,1265)
1265  FORMAT(/2X,16H PARAMETER NO. :)
      READ(NR,4040)AN
      J=AN
      ID=0
      IF(J.EQ.0)GO TO 110
      IF(J.LE.NEMAT(I,2))GO TO 1290
      ID=J-NEMAT(I,2)
      NPAR=NPAR+ID
      IF(I.GE.NEMAX)GO TO 1290
      K=NEMAT(I,1)+NEMAT(I,2)
      DO 1270 L=K,NPAR-ID
1270  EEN(L+ID)=EEN(L)
      DO 1275 L=1,NEMAX
      IF(NEMAT(L,1).GT.NEMAT(I,1))NEMAT(L,1)=NEMAT(L,1)+ID
1275  CONTINUE
1280  CONTINUE
      DO 1285 L=K,K+ID-1
1285  EEN(L)=0.0
1290  L=NEMAT(I,1)+J-1
      WRITE(NW,1295)EEN(L)
1295  FORMAT(2X,F15.4,3H  :)
      READ(NR,4040)EEN(L)
      NEMAT(I,2)=NEMAT(I,2)+ID
      GO TO 1260
C
C *** CHANGE UNIT
C
```

```
      CALL COPY8(A(1),A(3))
      IF(ICOMP.NE.1)GO TO 1306
1302  DO 1304 J=1,NEMAX
      CALL COMP8(A(1),ENAME(J),ICOMP)
      IF(ICOMP.EQ.1)GO TO 226
1304  CONTINUE
      ASSIGN 1302 TO KTYPE
      GO TO 1242
1306  WRITE(NW,1229)
      GO TO 212
C
C *** INPUT PARAMETERS
C
1310  CALL COPY8(A(1),A(4))
1320  DO 1330 I=1,NEMAX
      CALL COMP8(A(1),ENAME(I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1340
1330  CONTINUE
      ASSIGN 1320 TO KTYPE
      GO TO 1242
1340  KALL=10
      GO TO 307
C
C *** CHANGE COMPONENTS
C
1350  IF(NOCOMP.EQ.0)GO TO 500
      DO 1355 I=1,NOCOMP
1355  PROPRT(I,3)=0.0
      GO TO 500
C
C *** CHANGE  ALL
C
1400  DO 1405 I=1,7
      CALL COMP8(A(3),VNAMES(26+I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1410
1405  CONTINUE
      GO TO 1035
1410  KALL=11
      GO TO (3500,800,700,500,300,400,600),I
C
```

```
C *** TYPE
C
1500  NW=NW1
1501  DO 1505 I=1,10
      CALL COMP8(A(2),VNAMES(21+I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1510
1505  CONTINUE
      GO TO 1035
1510  KALL=12
      GO TO (1700,2521,1600,1515,1580,1650,1045,2435,1670,1550),I
C
C *** TYPE STREAM ---
C
1515  DO 1520 I=1,NSMAX
      CALL COMP8(A(N),SNAME(I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1525
1520  CONTINUE
      I=2
      GO TO 1222
1525  WRITE(NW,1530)A(N)
1530  FORMAT(/2X,7H STREAM  ,2X,A8/)
      J1=-1
      I1=(I-1)*NSLMAX
      DO 1535 J=1,NSLMAX
      J1=J1+2
1535  WRITE(NW,1540)VNAME(J1),VNAME(J1+1),SMATRX(I1+J)
1540  FORMAT(2X,2A8,2X,F15.5)
      GO TO 110
C
C *** TYPE PARAMETERS ---
C
1550  DO 1555 I=1,NEMAX
      CALL COMP8(A(N),ENAME(I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1560
1555  CONTINUE
      ASSIGN 1550 TO KTYPE
      GO TO 1242
1560  L=NEMAT(I,1)
      M=NEMAT(I,2)
      WRITE(NW,1565)A(N),M
1565  FORMAT(/2X,7H UNIT   ,A8,2X,5H HAS  ,I3,11H PARAMETERS   /)
      IF(M.EQ.0)GO TO 110
```

```
      GO TO 110
C
C *** TYPE ROUTINE ---
C
1580  DO 1585 I=1,NEMAX
      CALL COMP8(A(N),ENAME(I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1590
1585  CONTINUE
      ASSIGN 1580 TO KTYPE
      GO TO 1242
1590  WRITE(NW,1595)A(N),RNAME(I)
1595  FORMAT(/2X,6H UNIT ,A8,24H IS REPRESENTED BY MODEL,2X,A8)
      GO TO 110
C
C *** TYPE UNIT ---
C
1600  DO 1605 I=1,NEMAX
      CALL COMP8(A(N),ENAME(I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1610
1605  CONTINUE
      ASSIGN 1600 TO KTYPE
      GO TO 1242
1610  WRITE(NW,1615)A(N)
1615  FORMAT(/2X,6H UNIT ,A8/)
      DO 1620 J=2,NCOLPM
      JK=KPM(I,J)
      IF(JK.LE.0)GO TO 1630
      J1=J-1
1620  WRITE(NW,1625)J1,SNAME(JK)
1625  FORMAT(2X,18H INPUT  STREAM NO.,I2,2X,A8)
1630  DO 1635 K=J,NCOLPM
      JK=IABS(KPM(I,K))
      IF(JK.EQ.0)GO TO 110
      J1=K-J+1
1635  WRITE(NW,1640)J1,SNAME(JK)
1640  FORMAT(2X,18H OUTPUT STREAM NO.,I2,2X,A8)
      GO TO 110
C
C *** TYPE PREFERRED AND ITERATE STREAMS
C
```

```
1650  WRITE(NW,1655)
1655  FORMAT(//4X,30H PREFERRED AND ITERATE STREAMS /)
      DO 1665 I=1,NSMAX
      IF(KSFLAG(I).NE.2)GO TO 1665
      WRITE(NW,1530)SNAME(I)
      J=-1
      I1=(I-1)*NSLMAX
      DO 1660 K=1,NSLMAX
      J=J+2
1660  WRITE(NW,1540)VNAME(J),VNAME(J+1),SMATRX(I1+K)
1665  CONTINUE
      GO TO 110
C
C *** TYPE COMPONENTS
C
1670  WRITE(NW,1675)
1675  FORMAT(//4X,11H COMPONENTS  /)
      IF(NOCOMP.EQ.0)GO TO 1685
      IF(NCTOT.EQ.0)NCTOT=NOCOMP
      DO 1680 J=1,NCTOT
1680  WRITE(NW,1690)PROPRT(J,1),PROPRT(J,2)
      GO TO 110
1685  WRITE(NW,1695)
      GO TO 110
1690  FORMAT(2X,2A8)
1695  FORMAT(/2X,26H NO COMPONENTS READ IN YET /)
C
C *** TYPE ALL ***
C
1700  DO 1705 I=1,7
      CALL COMP8(A(3),VNAMES(26+I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1710
1705  CONTINUE
      GO TO 1035
1710  KALL=10
      GO TO (1650,1045,2435,1670,2316,2378,2180),I
C
C *** PRINT  -  PRINT ALL
C
1800  NW=NW2
      GO TO 1501
C
```

```
C
1850  WRITE(NW,1855)
1855  FORMAT(2X,'TYPE THE ERROR PERCENTAGE AFTER THE STREAM NAME'/)
      DO 1865 I=1,NSMAX
      WRITE(NW,1860)SNAME(I)
1860  FORMAT(2X,A8)
      READ(NR,4040)ERR(I)
      ERR(I)=ERR(I)/100.
1865  CONTINUE
      GO TO 110
C
C     DATA ECHO
C
2000  CONTINUE
2001  WRITE(NW1,2005)TITLE
2005  FORMAT(//////1X,10A8/1X,80(1H*)//)
      ASSIGN 2015 TO KALLER
      GO TO 3000
2015  WRITE(NW1,2020)
2020  FORMAT(//10X,15H PROCESS MATRIX/11X,14(1H*)/)
      DO 2110 I=1,NEMAX
      IF(KPM(I,1).EQ.0)GO TO 2110
      WRITE(NW1,2035)ENAME(I)
2035  FORMAT(/10X,13H UNIT NAME : ,A8)
      DO 2070 J=2,NCOLPM
      JK=KPM(I,J)
      IF(JK.LT.1)GO TO 2075
      J1=J-1
2060  WRITE(NW1,2065)J1,SNAME(JK)
2065  FORMAT(15X,18H INPUT  STREAM NO.,I2,5X,A8)
2070  CONTINUE
2075  DO 2105 K=J,15
      JK=IABS(KPM(I,K))
      IF(JK.LT.1)GO TO 2110
      K1=K-J+1
      WRITE(NW1,2100)K1,SNAME(JK)
2100  FORMAT(15X,18H OUTPUT STREAM NO.,I2,5X,A8)
2105  CONTINUE
2110  CONTINUE
2115  WRITE(NW1,2120)
```

```
2120  FORMAT(//10X,14H PLANT FEEDS :  /11X,13(1H*)/)
      DO 2140 I=1,NSMAX
      IF(KSEM(I,2).EQ.0.AND.KSEM(I,3).NE.0)
     *WRITE(NW1,2175)SNAME(I)
2140  CONTINUE
      WRITE(NW1,2150)
2150  FORMAT(//10X,16H PLANT OUTPUTS :  /11X,15(1H*)/)
      DO 2170 I=1,NSMAX
      IF(KSEM(I,3).EQ.0.AND.KSEM(I,2).NE.0)
     *WRITE(NW1,2175)SNAME(I)
2170  CONTINUE
2175  FORMAT(16X,A8)
2180  WRITE(NW1,2185)
2185  FORMAT(//10X,14H STREAM MATRIX/11X,13(1H*))
      NLM=NSMAX/3
      NM=NSMAX-NLM*3
      IF(NLM.LT.1)GO TO 2270
      I1=-2
      DO 2265 I=1,NLM
      I1=I1+3
      I2=I1+2
      WRITE(NW1,2230)(SNAME(K),K=I1,I2)
2230  FORMAT(//32X,3(7X,A8)/)
      J1=-1
      DO 2260 J=1,NSLMAX
      J1=J1+2
      WRITE(NW1,2255)VNAME(J1),VNAME(J1+1),(SMATRX((L-1)*NSLMAX+J)
     ,,L=I1,I2)
2255  FORMAT(11X,2A8,4X,3F15.5)
2260  CONTINUE
2265  CONTINUE
2270  IF(NM.EQ.0)GO TO 2315
      NLM=3*NLM+1
      WRITE(NW1,2230)(SNAME(K),K=NLM,NSMAX)
      J1=-1
      WRITE(NW1,2295)
2295  FORMAT(10X)
      DO 2310 J=1,NSLMAX
      J1=J1+2
2310  WRITE(NW1,2255)VNAME(J1),VNAME(J1+1),(SMATRX((L-1)*NSLMAX+J)
     ,,L=NLM,NSMAX)
2315  IF(KALL.GT.9)GO TO 110
```

```
      IF(NPAR.GT.0)GO TO 2330
      WRITE(NW1,2325)
 2325 FORMAT(/10X,50H ** NO PARAMETERS HAVE BEEN GIVEN FOR THE UNITS **
     */)
      GO TO 2375
 2330 DO 2360 I=1,NEMAX
      J=NEMAT(I,1)
      K=NEMAT(I,2)
      IF(K.EQ.0)GO TO 2350
      K=K+J-1
      WRITE(NW1,2340)ENAME(I),(EEN(L),L=J,K)
 2340 FORMAT(/10X,A8,2X,4F15.5,7(/20X,4F15.5))
      GO TO 2360
 2350 WRITE(NW1,2370)ENAME(I)
 2370 FORMAT(/10X,A8,5X,'HAS NO PARAMETERS')
 2360 CONTINUE
 2375 IF (KALL.GT.9)GO TO 110
 2378 WRITE(NW1,2380)
 2380 FORMAT(//10X,23H UNITS AND MODELS NAMES/11X,22(1H*)/)
      IF(IRT.GT.0)GO TO 2385
      WRITE(NW1,2386)
 2386 FORMAT(/10X,38H ** MODEL NAMES HAVE NOT BEEN GIVEN **  /)
      GO TO 2400
 2385 DO 2390 I=1,NEMAX
      CALL COMP8(RNAME(I),ZERO,ICOMP)
      IF(ICOMP.EQ.1)GO TO 2387
      WRITE(NW1,2395)ENAME(I),RNAME(I)
      GO TO 2390
 2387 WRITE(NW1,2388)ENAME(I)
 2388 FORMAT(10X,'UNIT',2X,A8,2X,' HAS NO MODEL ASSIGNED TO IT')
 2390 CONTINUE
 2395 FORMAT(10X,6H UNIT ,2X,A8,2X,24H REPRESENTED BY ROUTINE ,2X,A8)
 2400 IF(KALL.GT.9)GO TO 110
      WRITE(NW1,2405)
 2405 FORMAT(//10X,19H PROBLEM DIMENSIONS/11X,18(1H*)/)
      WRITE(NW1,2415)NEMAX,NSMAX,NOCOMP,NSLMAX
 2415 FORMAT(/10X,18H NUMBER OF UNITS = ,I8
     1/10X,20H NUMBER OF STREAMS = ,I6
     2/10X,24H NUMBER OF COMPONENTS = ,I2
     3/10X,17H STREAM LENGTH = ,I9
```

```
      *)
2435   IF(LOOPN.LT.1)GO TO 2480
       WRITE(NW1,2440)
2440   FORMAT(//10X,1(H PROBLEM CONTROLS/11X,16(1H*)/)
       WRITE(NW1,2450)NLOOPS,TOL
2450   FORMAT(10X,34H MAXIMUM NUMBER OF LOOPS ALLOWED = ,I5/
      110X,20H TOLERANCE ALLOWED = ,F10.5/)
       IF(KONVRG.GT.0)GO TO 2480
2465   WRITE(NW1,2470)
2470   FORMAT(10X,29H CONVERGENCE PROMOTER NOT SET)
       GO TO 2490
2480   WRITE(NW1,2485)KONVRG
2485   FORMAT(10X,25H CONVERGENCE PROMOTER NO.,I2,4H SET)
       GO TO 2490
2486   WRITE(NW1,775)
2490   IF(ITRACE)2510,2495,2497
2495   WRITE(NW1,2500)
       GO TO 2520
2497   WRITE(NW1,2498)ITRACE
2498   FORMAT(10X,17H TRACE SET EVERY ,I3,6H LOOPS/)
2500   FORMAT(10X,23H TRACE SET ON ALL UNITS /)
       GO TO 2520
2510   WRITE(NW1,2515)
2515   FORMAT(10X,21H TRACE OPTION NOT SET /)
2520   IF(KALL.GT.9)GO TO 110
2521   IF(NLIST.GT.0)GO TO 2540
       WRITE(NW1,2530)
2530   FORMAT(/10X,28H CALCULATION ORDER NOT GIVEN/11X,27(1H*)/)
       GO TO 2620
2540   WRITE(NW1,2545)
2545   FORMAT(/10X,18H CALCULATION ORDER/11X,17(1H*))
       ISEQ=2-NELIST(1,2)
       DO 2615 I=1,NLIST
       IS2=NELIST(I,2)/2
       IF(IS2-ISEQ)2570,2605,2590
2570   WRITE(NW1,2575)
2575   FORMAT(/12X,15H SEQUENTIAL SET/)
       ISEQ=0
       GO TO 2605
2590   WRITE(NW1,2595)
2595   FORMAT(/12X,13H RECYCLE LOOP/)
       ISEQ=1
```

```
 2615  CONTINUE
 2620  IF(KALL.EQ.12)GO TO 110
       WRITE(NW1,2625)
 2625  FORMAT(////30X,17H END OF DATA ECHO  /31X,16(1H*)///)
       IF(INW.EQ.1)GO TO 2630
       INW=1
       NW=NW1
       NW1=NW2
       GO TO 2001
 2630  NW1=NW
       INW=0
       GO TO 110
C
C *** CALCULATE  (STORE THE DATA)
C
 2700  KALL=12
       IERR=N-2
       IF(NPAR.GT.1)GO TO 2730
       WRITE(NW,2725)
 2725  FORMAT(/2X,16H *** WARNING ***/2X,39H *** PARAMETERS LISTS NOT YET
      T INPUT ***/)
       GO TO 300
 2730  IF(IRT.EQ.1)GO TO 2740
       WRITE(NW,2735)
 2735  FORMAT(/2X,16H *** WARNING *** /2X,35H *** MODELS NAMES NOT YET IN
      NPUT ***/)
       GO TO 400
 2740  IF(NCTOT.GT.0)GO TO 2750
       WRITE(NW,2745)
 2745  FORMAT(/2X,16H *** WARNING ***/2X,39H *** COMPONENTS NAMES NOT YET
      T INPUT ***/)
       GO TO 500
 2750  IF(NLIST.GT.0)GO TO 2800
       WRITE(NW,2755)
 2755  FORMAT(/2X,16H *** WARNING ***/2X,45H *** COMPUTATIONAL SEQUENCE N
      NOT YET INPUT ***/)
       GO TO 3300
 2800   WRITE(NW,2805)
 2805  FORMAT(/2X,60H *** ALL DATA AVAILABLE AND WILL BE STORED FOR LATER
```

```
     R USE ***/2X,46H *** THE EXECUTIVE PROGRAM IS BEING LOADED ***//)
      IF(NLIST.GT.0)GO TO 3040
      ASSIGN 3040 TO KALLER
C
C *** DATA LOGGER
C
C *** CREATING THE STREAM CONNECTION MATRIX
C
3000  DO 3005 I=1,NEMAX
      DO 3005 J=1,3
3005  KSEM(I,J)=0
      DO 3035 I=1,NEMAX
      IF(KPM(I,1).EQ.0)GO TO 3035
      J=KPM(I,1)
      DO 3030 K=2,NCOLPM
      IF(KPM(I,K))3015,3035,3010
3010  L=KPM(I,K)
      KSEM(L,3)=J
      GO TO 3025
3015  L=-KPM(I,K)
      KSEM(L,2)=J
3025  KSEM(L,1)=L
3030  CONTINUE
3035  CONTINUE
      IF(INDEX.EQ.3)GO TO 3040
      GO TO KALLER
C
C *** SENDING THE DATA TO THE LOG FILE
C
3040  PAUSE FILEA
      GO TO 3045
3045  WRITE(NWF,4000)TITLE
      WRITE(NWF,4010)NEMAX,NSMAX,NOCOMP,NSLMAX,NPAR,NCOLPM,KONVRG,
     *NPS,ITRACE,NLOOPS,NLIST,NVNAME,NPROP,NOGO,IUNITS,NUSRPPT,NCTOT
     *,NPROPX,IRT,IERR
      WRITE(NWF,4020)TOL
      WRITE(NWF,4000)(SNAME(I),I=1,NSMAX)
      WRITE(NWF,4000)(ENAME(I),I=1,NEMAX)
      IF(IRT.EQ.0)GO TO 3060
      WRITE(NWF,4000)(RNAME(I),I=1,NEMAX)
      WRITE(NWF,4010)(MODULE(I),I=1,NEMAX)
3060  WRITE(NWF,4000)(VNAME(I),I=1,NVNAME)
```

```
      WRITE(NWF,4000)PROPRT(I,1),PROPRT(I,2)
      WRITE(NWF,4020)(PROPRT(I,J),J=3,NPROP)
3070  WRITE(NWF,4020)(PRTEXTRA(I,J),J=1,NPROPX)
3080  IF(NUSRPPT.EQ.0)GO TO 3100
      DO 3090 I=1,NCTOT
3090  WRITE(NWF,4020)(USERPPT(I,J),J=1,NUSRPPT)
3100  WRITE(NWF,4010)((KPM(I,J),J=1,NCOLPM),I=1,NEMAX)
      WRITE(NWF,4010)((KSEM(I,J),J=1,3),I=1,NSMAX)
3110  WRITE(NWF,4020)(SMATRX(I),I=1,NSLMAX*NSMAX)
3120  IF(NPAR.LT.1)GO TO 3130
      WRITE(NWF,4020)(FEN(I),I=1,NPAR)
      WRITE(NWF,4010)((NEMAT(I,J),J=1,2),I=1,NEMAX)
3130  IF(NLIST.EQ.0)GO TO 3140
      WRITE(NWF,4010)((NELIST(I,J),J=1,2),I=1,NLIST)
      WRITE(NWF,4010)(KSFLAG(I),I=1,NSMAX)
3140  IF(NPS.EQ.0)GO TO 3145
      WRITE(NWF,4010)(KPS(I),I=1,NPS)
3145  IF(IERR.EQ.0)GO TO 3150
      WRITE(NWF,4020)(ERR(I),I=1,NSMAX)
C
C
3150  IF(INDEX.EQ. 3)GO TO 3160
      IRETURN=-1
      IF(JSWTCH2.EQ.1.AND.IPM.EQ.1)RETURN
      STOP
3160  PAUSE EXIT
C
C
C
C *** USER-PROVIDED CALCULATION ORDER
C
3300  WRITE(NW,3305)
3305  FORMAT(/2X,50H DO YOU WISH TO INPUT YOUR OWN CALCULATION ORDER ?
     *)
      ASSIGN 3310 TO KALLER
      READ(NR,4000)BUFFER
      GO TO 1000
3310  IF(NO.EQ.1)GO TO 3500
3312  WRITE(NW,3315)
3315  FORMAT(/2X,8H ORDER :/)
```

```
3320  READ(NR,4000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
3325  I=4
      CALL COMP(I,A(1),1,VNAMES(3),1)
      IF(I.NE.4)GO TO 3330
      ISEQ=2
      ISQ=1
      GO TO 3335
3330  I=3
      CALL COMP(I,A(1),1,VNAMES(4),1)
      IF(I.NE.3)GO TO 3335
      ISEQ=1
      ISQ=1
      IF(NNE.LT.2)GO TO 3340
3335  NELIST(NNE-1,2)=3
3340  DO 3365 J=1,N
      IF(ISQ.NE.1)GO TO 3345
      ISQ=0
      GO TO 3365
3345  DO 3350 K=1,NEMAX
      CALL COMP8(A(J),ENAME(K),ICOMP)
      IF(ICOMP.EQ.1)GO TO 3360
3350  CONTINUE
      WRITE(NW,3355)A(J)
3355  FORMAT(/2X,A8,2X,30H IS NOT A RECOGNISED UNIT NAME /
     *2X,40H PLEASE RETYPE JUST THIS UNIT NAME AGAIN/)
      READ(NR,4000)BUFFER
      M=1
      CALL COPYER(8,B,BUFFER,M)
      CALL COPY8(A(J),B(1))
      J=J-1
      GO TO 3365
3360  NELIST(NNE,1)=K
      NELIST(NNE,2)=ISEQ
      NNE=NNE+1
3365  CONTINUE
      NLIST=NNE-1
      IF(NLIST.LT.NEMAX)GO TO 3320
      IF(NELIST(NLIST,2).EQ.2)NELIST(NLIST,2)=3
      WRITE(NW,3370)
3370  FORMAT(/2X,18H ITERATE STREAMS : / 2X,11H HOW MANY ?/)
```

```
      IF(NPS.EQ.0)GO TO 3405
      WRITE(NW,3375)
 3375 FORMAT(2X,18H ITERATE STREAMS :/)
      K=0
 3380 READ(NR,4000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
      IF(N.EQ.0)GO TO 3380
      DO 3400 I=1,N
      DO 3385 J=1,NSMAX
      CALL COMP8(SNAME(J),A(I),ICOMP)
      IF(ICOMP.EQ.1)GO TO  3395
 3385 CONTINUE
      WRITE(NW,3390)A(I)
 3390 FORMAT(/2X,A8,2X,32H IS NOT A RECOGNISED STREAM NAME /
     *2X,42H PLEASE RETYPE JUST THIS STREAM NAME AGAIN /)
      READ(NR,4000)BUFFER
      M=1
      CALL COPYER(8,B,BUFFER,M)
      CALL COPY8(A(I),B(1))
      I=I-1
      GO TO 3400
 3395 KPS(K+I)=J
      KSFLAG(I)=2
 3400 CONTINUE
      K=K+N
      IF(K.LT.NPS)GO TO 3380
      ASSIGN 300 TO KEXIT
 3405 IF(KALL.EQ.12)GO TO 2800
      IF(KALL.GT.9)GO TO 110
      GO TO 1020
C
C *** USER DOES NOT PROVIDE THE CALCULATION ORDER
C
 3500 ASSIGN 3510 TO KALLER
      GO TO 3000
 3510 DO 3520 I=1,NSMAX
      IF(KSEM(I,2).GT.0)GO TO 3520
      KSFLAG(I)=1
 3520 CONTINUE
```

```
      IRETURN=1
      RETURN
3530  IF(KALL.EQ.12)GO TO 2800
      IF(KALL.GT.9)GO TO 110
      ASSIGN 300 TO KEXIT
      GO TO 1020
C
C *** FORMATS
C
4000  FORMAT(10A8)
4010  FORMAT(16I5)
4020  FORMAT(5E15.7)
4030  FORMAT(I0)
4040  FORMAT(F0.0)
4050  FORMAT(1E15.7)
4060  FORMAT(30F0.0)
4500  FORMAT(//20X,27H ****  PEETPACK MK. 1  ****  /)
4510  FORMAT(/4X,19H CASE STUDY TITLE : )
C
      END
      SUBROUTINE COPYER(M,A,B,NCOPY)
C
      DIMENSION FORMS(2),FORM(3)
      DIMENSION A(20),B(10),SP(2)
      DATA FORM/8H(       ,8H        ,8HX,1F0.0)/
      DATA SP/8H        ,8H        /
C
      CALL COPY8(A(1),SP(1))
      IF(NCOPY+1)1,2,3
1     NC=-1
      NCOPY=-NCOPY-1
      GO TO 5
2     NCOPY=81
3     NC=0
5     K=1
      N=1
      DO 60 I=1,80
      I=1
      CALL COMP(J,B(1),I,SP(1),1)
      IF(J)10,10,20
10    CALL COPY(1,A(N),K,B(1),I)
      IF(K.GE.M)GO TO 50
```

```
30     I=M-K+1
       IF(L)50,50,40
40     CALL COPY(L,A(N),K,SP(1),1)
50     K=1
       N=N+M/8
       IF(N-NCOPY)60,60,70
60     CONTINUE
70     IF(NC.GE.0)GO TO 110
       I=I-1
       WRITE(8,100)I
100    FORMAT(I8)
       READ(8,101)FORM(2)
101    FORMAT(A8)
       READ(7,FORM)A(NCOPY+1)
110    NCOPY=N-1
       RETURN
       END
       SUBROUTINE ORDER(NR,NW)
C
       DIMENSION LOOP(20,30),IP(30),ISTBUF(30),VNAMES(2),BUFFER(10),
      *A(20),B(20)
       COMMON
      *KPM(30,8),KSFLAG(30),KSEM(30,3),NELIST(30,2),KPS(10),
      *NEMAX,NSMAX,NCOLPM,NLIST,NPS,NLOOPS,ENAME(30),SNAME(30)
       COMMON /ORDR/ IH(30,30),KFLAG1,LOOPN,NE1,IE,KEFLAG(30)
       DATA VNAMES/8HYES     ,8HNO      /
C
       DO 5 K=1,NSMAX
       KSFLAG(K)=0
       I=KSEM(K,2)
       J=KSEM(K,3)
5      ISTBUF(K)=I*100+J
       DO 20 I=1,NEMAX
       IP(I)=0
       DO 10 J=1,20
10     LOOP(J,I)=0
       DO 20 J=1,NEMAX
       IH(I,J)=0
20     CONTINUE
```

```
C
C      ***  EC 1  ***
C
       NLOOPS=0
       K=1
       IP(1)=1
C
C      ***  EC 2  ***
C
30     N=IP(K)
       DO 50 J=2,NCOLPM
       L=-KPM(N,J)
       IF(L.EQ.0)GO TO 70
       IF(L.LT.0)GO TO 50
       L=KSEM(L,3)
       IF(L.LE.IP(1))GO TO 50
       DO 40 M=1,NEMAX
       IF(L.EQ.IP(M))GO TO 50
       I=IP(K)
       IF(L.EQ.IH(I,M))GO TO 50
40     CONTINUE
       GO TO 60
50     CONTINUE
       GO TO 70
60     K=K+1
       IP(K)=L
       GO TO 30
C
C      ***  EC 3  ***
C
70     N=IP(K)
       DO 80 J=2,NCOLPM
       L=-KPM(N,J)
       IF(L.EQ.0)GO TO 110
       IF(L.LT.0)GO TO 80
       L=KSEM(L,3)
       IF(IP(1).EQ.L)GO TO 90
80     CONTINUE
       GO TO 110
C
90     NLOOPS=NLOOPS+1
       DO 100 J=1,NEMAX
```

```
C
110   IF(K.EQ.1)GO TO 150
      L=IP(K)
      DO 120 M=1,NEMAX
120   IH(L,M)=0
      L=IP(K-1)
      DO 130 M=1,NEMAX
      IF(IH(L,M).EQ.0)GO TO 140
130   CONTINUE
C
140   IH(L,M )=IP(K)
      IP(K)=0
      K=K-1
      GO TO 30
C
C     ***  EC 5  ***
C
150   IF(IP(1).GE.NEMAX)GO TO 180
      IP(1)=IP(1)+1
      K=1
      DO 160 I=1,NEMAX
      DO 160 J=1,NEMAX
160   IH(I,J)=0
      GO TO 30
C
180   IF(NLOOPS.GT.0)GO TO 185
      WRITE(NW,184)
184   FORMAT(2X,31H PLANT FLOW DIAGRAM SEQUENTIAL /
     *2X,28H NO ITERATE STREAMS REQUIRED  //)
      GO TO 1500
185   DO 190 I=1,NEMAX
      DO 190 J=1,NEMAX
190   IH(I,J)=0
200   DO 270 L=1,NLOOPS
      DO 220  I=1,NEMAX
      IF(LOOP(L,I).GT.0)GO TO 210
      IP(I)=IP(1)
      GO TO 230
210   IP(I)=LOOP(L,I)
```

```
      LOOP(L,I)=0
220   CONTINUE
230   I=I-1
      DO 260 J=1,I
      J1=IP(J)
      IH(L,J1)=1
      J2=IP(J+1)
      IST=J1*100+J2
      IP(J)=0
      DO 240 K=1,NSMAX
      IF(ISTBUF(K).EQ.IST)GO TO 250
240   CONTINUE
250   LOOP(L,K)=1
260   CONTINUE
      IP(I+1)=0
270   CONTINUE
C
C *** CUMULATIVE MATRIX
C
      NLCUML=NLOOPS
      N=0
      DO 450 I=1,NLCUML
400   DO 440 J=1,NEMAX
      IF(IH(I,J).LT.1)GO TO 440
      IF(I.GE.NLCUML)GO TO 460
      DO 430 K=I+1,NLCUML
      IF(IH(K,J).LT.1)GO TO 430
      DO 420 L=1,NEMAX
      IH(I,L)=IH(I,L)+IH(K,L)
      IH(K,L)=0
      IF(K.EQ.NLCUML)GO TO 420
      DO 410 M=K+1,NLCUML
410   IH(M-1,L)=IH(M,L)
420   CONTINUE
425   N=1
      NLCUML=NLCUML-1
430   CONTINUE
440   CONTINUE
      IF(N.EQ.0)GO TO 450
      N=0
      GO TO 400
450   CONTINUE
```

```
      IF(IH(I,J).GT.0)IH(I,NEMAX+1)=IH(I,NEMAX+1)+1
500   CONTINUE
C
C *** SEARCH FOR ITERATE STREAMS TO CUT ALL LOOPS
C
1000  NS=0
      N1=-1
C *** GET NUMBER OF LOOPS IN WHICH EACH STREAM IS INVOLVED
      DO 1105 I=1,NSMAX
      LOOP(NLOOPS+1,I)=0
      DO 1100 L=1,NLOOPS
      IF(LOOP(L,I).EQ.2)LOOP(L,I)=1
1100  LOOP(NLOOPS+1,I)=LOOP(NLOOPS+1,I)+LOOP(L,I)
1105  CONTINUE
C
C *** FLAG PREFERRED STREAMS  100  IN ROW NLOOPS+1 OF ITERATE STREAM SEARCH
C *** MATRIX
C
      IF(NPS)1130,1130,1110
1110  DO 1120 I=1,NPS
      J=KPS(I)
1120  LOOP(NLOOPS+1,J)=NLOOPS+1
1130  N1=N1+1
      DO 1230 I=1,NSMAX
      JJ=0
      IF(LOOP(NLOOPS+1,I)-N1)1230,1140,1230
1140  NS=NS+1
      IF(N1)1230,1230,1150
1150  DO 1200 J=1,NLOOPS
      IF(LOOP(J,I))1200,1200,1160
1160  DO 1190 K=1,NSMAX
      IF(K-I)1170,1190,1170
1170  IF(LOOP(J,K).NE.1)GO TO 1190
1180  LOOP(J,I)=2
      JJ=JJ+1
      GO TO 1200
1190  CONTINUE
      GO TO 1220
1200  CONTINUE
```

```
      IF(JJ-N1)1220,1210,1210
1210  LOOP(NLOOPS+1,I)=0
      GO TO 1230
1220  LOOP(NLOOPS+1,I)=100
      DO 1225 J=1,NLOOPS
      IF(LOOP(J,I).EQ.2)LOOP(J,I)=1
1225   CONTINUE
1230  CONTINUE
      IF(NS.LT.NSMAX)GO TO 1150
C
C *** LIST PREFERRED STREAMS
C
1235  WRITE(NW,1240)
1240  FORMAT(/2X,28H LIST OF PREFERRED STREAMS :)
      NPS=0
      DO 1250 I=1,NSMAX
      IF(LOOP(NLOOPS+1,I).LT.1)GO TO 1250
      WRITE(NW,1245)SNAME(I)
1245  FORMAT(2X,A8)
1250  CONTINUE
C
C *** CHECK WITH USER FOR ACCEPTANCE OF PREFERRED STREAMS
C
      WRITE(NW,1255)
1255  FORMAT(/2X,21H DO YOU ACCEPT THEM ?)
1260  READ(NR,1265)BUFFER
1265  FORMAT(10A8)
      N=1
      CALL COPYER(8,A,BUFFER,N)
      CALL COMP8(A(1),VNAMES(1),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1360
      CALL COMP8(A(1),VNAMES(2),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1295
      WRITE(NW,1270)
1270  FORMAT(/2X,32H PLEASE RETYPE YOUR ANSWER AGAIN /)
      GO TO 1260
1295  WRITE(NW,1300)
1300  FORMAT(/2X,50H HOW MANY PREFERRED STREAMS DO YOU WISH TO INPUT ?)
      READ(NR,1305)PS
1305  FORMAT(F0.0)
      NPS=PS
```

```
      IF(NPS.EQ.0)GO TO 1360
      WRITE(NW,1310)
 1310 FORMAT(/2X,34H NAMES OF YOUR PREFERRED STREAMS : )
      K=0
 1315 READ(NR,1265)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
      IF(N.EQ.0)GO TO 1315
      DO 1350 I=1,N
      DO 1320  J=1,NSMAX
      CALL COMP8(A(I),SNAME(J),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1345
 1320 CONTINUE
      WRITE(NW,1325)A(I)
 1325 FORMAT(/2X,A8,2X,32H IS NOT A RECOGNISED STREAM NAME /
     *2X,42H PLEASE RETYPE JUST THIS STREAM NAME AGAIN /)
      M=1
      READ(NR,1265)BUFFER
      CALL COPYER(8,B,BUFFER,M)
      CALL COPY8(A(I),B(1))
      I=I-1
      GO TO 1350
 1345 KPS(K+I)=J
 1350 CONTINUE
      K=K+N
      IF(K.LT.NPS)GO TO 1315
      GO TO 1000
 1360 DO 1370 J=1,NSMAX
      IF(KSEM(J,2).GT.0)GO TO 1365
      KSFLAG(J)=1
      GO TO 1370
 1365 IF(LOOP(NLOOPS+1,J).EQ.0)GO TO 1370
      NPS=NPS+1
      KPS(NPS)=J
      KSFLAG(J)=2
 1370 CONTINUE
C
C *** OBTAIN THE CALCULATION ORDER
C
```

```
      DO 1505 IE=1,NEMAX
 1505 KEFLAG(IE)=0
 1500 KFLAG=0
      DO 1600 IE=1,NEMAX
      IF(KEFLAG(IE).GT.0)GO TO 1600
      DO 1510 I=2,NCOLPM
      J=KPM(IE,I)
      IF(J.LE.0)GO TO 1520
      IF(KSFLAG(J).EQ.0)GO TO 1600
 1510  CONTINUE
 1520 IF(NLOOPS.LT.1)GO TO 1570
      DO 1550 LOOPN=1,NLCUML
      IF(IH(LOOPN,IE).GT.0)GO TO 1560
 1550 CONTINUE
      GO TO 1570
 1560 KFLAG1=0
      CALL FLAGER
      IF(KFLAG1.EQ.1)KFLAG=1
      GO TO 1600
 1570 DO 1580 I1=I,NCOLPM
      J=-KPM(IE,I1)
      IF(J.LE.0)GO TO 1590
 1580 KSFLAG(J)=1
 1590 KFLAG=1
      NE1=NE1+1
      NELIST(NE1,1)=IE
      NELIST(NE1,2)=1
      KEFLAG(IE)=1
 1600 CONTINUE
      IF(KFLAG.EQ.1)GO TO 1500
C     WRITE(2,2000)((NELIST(I,J),J=1,2),I=1,NEMAX),NE1
 2000 FORMAT(4X,6H ORDER ,12I5)
      NLIST=NEMAX
      RETURN
      END
      SUBROUTINE FLAGER
C
      COMMON /ORDR/ IH(30,30),KFLAG1,LOOPN,NE1,IE,KEFLAG(30)
      COMMON
     *KPM(30,8),KSFLAG(30),KSEM(30,3),NELIST(30,2),KPS(10),
     *NEMAX,NSMAX,NCOLPM,NLIST,NPS,NLOOPS,ENAME(30),SNAME(30)
```

```
      NEL=IH(LOOPN,NEMAX+1)
      K1=IE-1
20    DO 200 I=1,NEMAX
30    K1=K1+1
      IF(K1.GT.NEMAX)K1=1
      IF(IH(LOOPN,K1).LE.0)GO TO 30
40    IF(KEFLAG(K1).GT.0)GO TO 200
      DO 50 J=2,NCOLPM
      K3=KPM(K1,J)
      IF(K3.LE.0)GO TO 60
      IF(KSFLAG(K3).LE.0)GO TO 200
50    CONTINUE
60    KEFLAG(K1)=3
      NE1=NE1+1
      NELIST(NE1,1)=K1
      NELIST(NE1,2)=2
      NEFL=NEFL+1
      KFLAG=1
      DO 70 J1=J,NCOLPM
      K3=-KPM(K1,J1)
      IF(K3.EQ.0)GO TO 200
      IF(KSFLAG(K3).EQ.0)KSFLAG(K3)=3
70    CONTINUE
200   CONTINUE
      IF(KFLAG.EQ.0)GO TO 210
      KFLAG=0
      GO TO 20
210   IF(NEFL.NE.NEL)GO TO 250
      KFLAG1=1
      DO 220 I=1,NSMAX
      IF(KSFLAG(I).EQ.3)KSFLAG(I)=1
220   CONTINUE
      DO 230 I=1,NEMAX
      IF(KEFLAG(I).EQ.3)KEFLAG(I)=1
230   CONTINUE
      NELIST(NE1,2)=3
      RETURN
250   DO 260 I=1,NEMAX
      IF(KEFLAG(I).EQ.3)KEFLAG(I)=0
260   CONTINUE
```

```
      DO 270 I=1,NSMAX
      IF(KSFLAG(I).EQ.3)KSFLAG(I)=0
270   CONTINUE
      KFLAG1=0
      NE1=NE1-NEFL
      RETURN
      END
      SUBROUTINE EQCALLER(NWF)
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
      COMMON /CNAME/ SNAME(30),ENAME(30),RNAME(30),VNAME(60),
     1MODULE(30),TITLE(10),NSMAX,NEMAX,NPAR,NVNAME,ITRACE,NPROP,NUSRPPT
      PAUSE EQCAL
      WRITE(NWF,10)
10    FORMAT(6X,21HSUBROUTINE EQCALL(NE)/
     *6X,7HGO TO ()
      DO 20 I=1,NEMAX-1
20    WRITE(NWF,30)I
30    FORMAT(5X,1H*,I2,1H,)
      WRITE(NWF,40)
40    FORMAT(5X,1H*,I2,4H),NE)
      DO 50 I=1,NEMAX
50    WRITE(NWF,60)I,RNAME(I)
60    FORMAT(1X,I2,3X,5HCALL ,A8/6X,6HRETURN)
      WRITE(NWF,70)
70    FORMAT(6X,3HEND)
      RETURN
      END
      FINISH
```

Program BATCHER

```
      LIST (LP)
      PROGRAM (UPDATA)
      INPUT 1 = CR0
      INPUT 3 = CR1
      INPUT 5 = CR2
      INPUT 6 = CR3
      INPUT 9 = CR4
      INPUT 10 = CR5
      INPUT 11 = CR6
      OUTPUT 2 = LP0
      OUTPUT 4 = CP0
      TRACE 2
      USE 7 = /ARRAY
      USE 8 = /ARRAY
      END
      BLOCK DATA
C
      COMMON /DIMS1/
     *TITLE(10),BUFFER(10),VNAMES(31),A(20),
     *EEN(500),NEMAT(30,2),SMATRX(400),RNAME(30),VNAME(30),MODULE(30),
     *PROPRT(10,25),PRTEXTRA(10,35),USERPPT(10,10),SMODN(20),FORMS(2)
     *,VNAME2(6),ERR(30),ZERO
C
      COMMON /DIMS2/
     * NOCOMP,NSLMAX,NPAR,KONVRG,NPROPX
     *,ITRACE,NLOOPS,NVNAME,NPROP,NOGO,IUNITS
     *,NUSRPPT,NCTOT,IRT,IERR
C
      COMMON
     *KPM(30,8),KSFLAG(30),KSEM(30,3),NELIST(30,2),KPS(10),
     *NEMAX,NSMAX,NCOLPM,NLIST,NPS,ENAME(30),SNAME(30),TOL
C
      DATA VNAMES /
     * 4HLOOP,3HSEQ,
     * 8HEND     ,8HPROCESS ,8HPARAMETE,8HCOMPONEN,8HLIBRARY
     *,8HNEW     ,8HVARIABLE,8HKEEPON  ,8HNOGO    ,8H
     *,8HADDITION
     *,8HROUTINES,8HSTREAMS ,8HPREFERRE,8HORDER   ,8HNOORDER
     *,8HCALCULAT,8HLENGTH  ,8HBRITISH ,8HS.I.    ,8HACCELERA
     *,8HNOACCELE,8HTOLERANC,8HLOOPS   ,8HTRACE   ,8HNOTRACE
     *,8HDATA    ,8HEXIT    ,8HERRORS
     */
```

```
     *80HFLOW   (MOLAR)  TEMPERATURE      PRESSURE  ABS.  ENTHALPY
     1 VAPOUR FRACTION  /
C
      DATA VNAME2(1) /
     *48HURE DEGF  PSIA  BTU/LBMLURE DEGC  BARS   KJ/KGML  /
C
      DATA SMODN /
     *8HMASSMIX ,8HMIXER   ,8HSPLITTER,8HSPLITMIX,8HFLASHER ,
     *8HPURIFYER,8HSETBP   ,8HSETDP   ,8HHEATER  ,8HHEATEXCH,
     *8HPUMP    ,8HCOMPRESR,8HVALVE   ,8HDISCOL  ,8HREACTOR ,
     *8HUNIT1   ,8HUNIT2   ,8HUNIT3   ,8HUNIT4   ,8HUNIT5
     */
C
      DATA ZERO/8H00000000/
C
      END
      MASTER UPDATA
C
      COMMON /DIMS1/
     *TITLE(10),BUFFER(10),VNAMES(31),A(20),
     *EEN(500),NEMAT(30,2),SMATRX(400),RNAME(30),VNAME(30),MODULE(30),
     *PROPRT(10,25),PRTEXTRA(10,35),USERPPT(10,10),SMODN(20),FORMS(2)
     *,VNAME2(6),ERR(30),ZERO
C
      COMMON /DIMS2/
     * NOCOMP,NSLMAX,NPAR,KONVRG,NPROPX
     *,ITRACE,NLOOPS,NVNAME,NPROP,NOGO,IUNITS
     *,NUSRPPT,NCTOT,IRT,IERR
C
      COMMON
     *KPM(30,8),KSFLAG(30),KSEM(30,3),NELIST(30,2),KPS(10),
     *NEMAX,NSMAX,NCOLPM,NLIST,NPS,ENAME(30),SNAME(30),TOL
C
      COMMON /SWITCH/ JSWTCH
C
      EXTERNAL CHECKERR
C
C
C     INITIAL VALUES AND FLAGS
C
```

```
C
      NCOMND=29
C
      NEMAX=0
      NSMAX=0
      NSLMAX=5
      NL=5
      NOCOMP=0
      NCTOT=0
      NCMORE=0
      NPAR=0
      NPS=0
      NLIST=0
      NVNAME=10
      NLIST=0
      KONVRG=0
      ITRACE=-1
      INDEX=0
      IUNITS=0
      NCOLPM=0
      NCOLMX=1
      NIE=0
      NOGO=1
      ISEQ=1
      NNE=0
      ISQ=0
      IRT=0
      IPM=0
      IVN=0
      IERR=0
C
      DO 3 I=1,30
      DO 1 J=1,8
1     KPM(I,J)=0
      DO 2 J=1,2
2     NEMAT(I,J)=0
3     CONTINUE
      DO 4 I=1,500
4     EEN(I)=0.
C
C     ASSIGNING THE I/O PERIPHERALS
C
```

```
      NW=2
      NWUSER=2
      NRCOMPX=10
      NRCOMPX2=10
      NRCOMPX1=11
      NRCOMP=3
      NWFILE=4
      NRFILE=5
      NRCOMP1=6
      NRCOMP2=9
1000  FORMAT(10A8)
C
C     SETTING THE BUFFER AREAS AND THE ERROR TRAP
C
      J=1
      CALL SSWTCH(J,JSWTCH)
      J=2
      CALL SSWTCH(J,JSWTCH2)
      CALL DEFBUF(7,80,BUFFER)
      CALL DEFBUF(8,16,FORMS)
C
C     GO TO READ THE INPUT DATA
C
      IF(JSWTCH.EQ.1)GO TO 2000
C
C
      READ(NR,1000)TITLE
C
C     READ DATA INPUT COMMANDS
C
10    CALL COPY8(A(1),VNAMES(12))
      READ(NR,1000)BUFFER
      WRITE(NWUSER,1000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
      DO 40 I=1,NCOMND
      CALL COMP8(A(1),VNAMES(I+2),ICOMP)
      IF(ICOMP-1)40,20,40
20    IF(I-10)30,10,30
30    INDEX=I
      GO TO 70
```

```
40     CONTINUE
       IF(INDEX)80,80,60
60     GO TO (
      * 10,101,152,205,205,
      *225,255,280,285, 10,
      *237,
      *310,352,455,501,550,
      *600,650,670,680,700,
      *710,720,750,770,790,
      *800,900,1005
      *),INDEX
70     GO TO (
      * 10,100,150,200,200,
      *220,250,280,285, 10,
      *235,
      *300,350,450,500,550,
      *600,650,670,680,700,
      *710,720,750,770,790,
      *800,900,1001
      *),INDEX
80     WRITE(NWUSER,90)BUFFER
90     FORMAT(///10X,39H ERROR : UNKNOWN COMMAND IN DATA CARD =//
      11X,10A8,//30X,22H SIMULATION ABANDONNED//)
       PAUSE ER
C
C
C      PROCESS MATRIX CHANGES
C
100    IF(JSWTCH.EQ.2)GO TO 2100
       READ(NR,1000)BUFFER
       N=-1
       CALL COPYER(8,A,BUFFER,N)
101    IPM=1
       IF(JSWTCH.EQ.2)GO TO 2101
       DO 104 IE=1,NEMAX
       CALL COMP8(ENAME(IE),A(1),ICOMP)
       IF(ICOMP-1)104,112,104
104    CONTINUE
       IF(N-1)106,106,110
106    WRITE(NW,108)A(1)
108    FORMAT(//10X,21H ERROR IN UNIT NAME  ,A8/
      110X,37H ATTEMPT TO DELETE AN INEXISTING UNIT//)
```

```
      IE=NEMAX
      KPM(IE,1)=IE
      CALL COPY8(ENAME(NEMAX),A(1))
      GO TO 114
112   IF(N-1)132,132,114
114   DO 120 I=2,N
      DO 116 JS=1,NSMAX
      CALL COMP8(SNAME(JS),A(I),ICOMP)
      IF(ICOMP-1)116,118,116
116   CONTINUE
      NSMAX=NSMAX+1
      JS=NSMAX
      CALL COPY8(SNAME(NSMAX),A(I))
118   KPM(IE,I)=JS
      NCOLMX=NCOLMX+1
120   CONTINUE
      READ(NR,1000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
      IF(N.EQ.0)GO TO 127
      DO 126 I=1,N
      DO 122 JS=1,NSMAX
      CALL COMP8(SNAME(JS),A(I),ICOMP)
      IF(ICOMP-1)122,124,122
122   CONTINUE
      NSMAX=NSMAX+1
      JS= NSMAX
      CALL COPY8(SNAME(NSMAX),A(I))
124   NCOLMX=NCOLMX+1
      KPM(IE,NCOLMX)=-JS
126   CONTINUE
127   IF(NCOLMX-NCOLPM)130,130,128
128   NCOLPM=NCOLMX
130   NCOLMX=1
      NLIST=0
      GO TO 10
C
C     DELETE UNIT FROM ALL ARRAYS
C
132   IF(IE-NEMAX)136,134,136
```

```
134   N=NEMAT(IE,2)
      GO TO 149
136   M=NEMAT(IE,1)
      N=NEMAT(IE,2)
      DO 142 K=IE+1,NEMAX
      MODULE(K-1)=MODULE(K)
      CALL COPY8(RNAME(K-1),RNAME(K))
      CALL COPY8(ENAME(K-1),ENAME(K))
      DO 138 L=2,NCOLPM
138   KPM(K-1,L)=KPM(K,L)
      DO 140 L=1,2
140   NEMAT(K-1,L)=NEMAT(K,L)
142   CONTINUE
      DO 144 I= M+N,NPAR
144   EEN(I-N)=EEN(I)
146   DO 148 I=1,NEMAX-1
      IF(NEMAT(I,1).GT.M)NEMAT(I,1)=NEMAT(I,1)-N
148   CONTINUE
149   NPAR=NPAR-N
      NEMAX=NEMAX-1
      GO TO 10
C
C     EQUIPMENT PARAMETERS
C
150   IF(JSWTCH.EQ.2)GO TO 2150
      READ(NR,1000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
152   IF(JSWTCH.EQ.2)GO TO 2152
      DO 154 IE=1,NEMAX
      CALL COMP8(A(1),ENAME(IE),ICOMP)
      IF(ICOMP-1)154,158,154
154   CONTINUE
      IE=NIE
      IF(IE)155,155,185
155   WRITE(NWUSER,156)BUFFER
156   FORMAT(///10X,25H ERROR IN PARAMETER ARRAY /
     11X,10A8/10X,25H UNIT NAME NOT RECOGNISED  ///)
      PAUSE ER
158   N1=N-1
      IF(NEMAT(IE,1).GT.0)GO TO 159
      NEMAT(IE,1)=NPAR+1
```

```
      IDIFF=N
      GO TO 170
159   IDIFF=N1-NEMAT(IE,2)
      DO 160 I=1,NEMAX
      IF(NEMAT(I,1).GT.NEMAT(IE,1))NEMAT(I,1)=NEMAT(I,1)+IDIFF
160   CONTINUE
      J=NEMAT(IE,1)+NEMAT(IE,2)
      NEMAT(IE,2)=N1
      IF(IDIFF)162,170,166
162   DO 164 I=J,NPAR
164   EEN(I+IDIFF)=EEN(I)
      GO TO 170
166   IF(NPAR.EQ.0)GO TO 170
      DO 168 I=J,NPAR
      II=NPAR+J-I
168   EEN(II+IDIFF)=EEN(II)
170   IF(N1.LT.1)GO TO 184
      DO 172 I=1,8
      J=1
      CALL COMP(J,A(1),I,VNAMES(12),1)
      IF(J)172,172,174
172   CONTINUE
174   K=I/2
      N1=N1-1
      GO TO(176,178,180,182),K
176   READ(7,177)(EEN(I),I=NEMAT(IE,1),NEMAT(IE,1)+N1)
177   FORMAT(2X,30F0.0)
      GO TO 184
178   READ(7,179)(EEN(I),I=NEMAT(IE,1),NEMAT(IE,1)+N1)
179   FORMAT(4X,30F0.0)
      GO TO 184
180   READ(7,181)(EEN(I),I=NEMAT(IE,1),NEMAT(IE,1)+N1)
181   FORMAT(6X,30F0.0)
      GO TO 184
182   READ(7,183)(EEN(I),I=NEMAT(IE,1),NEMAT(IE,1)+N1)
183   FORMAT(8X,30F0.0)
184   NIE=IE
      N1=N1+1
      NPAR=NPAR+IDIFF
      GO TO 10
185   IDIFF=N
```

```
      IE=NIE
      DO 186 I=1,NEMAX
      IF(NEMAT(I,1).GT.NEMAT(IE,1))NEMAT(I,1)=NEMAT(I,1)+IDIFF
186   CONTINUE
      J=NEMAT(IE,1)+NEMAT(IE,2)
      DO 187 I=J,NPAR
      II=NPAR+J-I
187   EEN(II+IDIFF)=EEN(II)
      READ(7,188)(EEN(II),II=J,J+N-1)
188   FORMAT(30F0.0)
      NEMAT(IE,2)=NEMAT(IE,2)+N
      NPAR=NPAR+N
      GO TO 10
C
C     LIBRARY COMPONENT NAMES
C
200   READ(NR,1000)BUFFER
      NOCOMP=0
      NCTOT=0
      NVNAME=NL+NL
      NSLMAX=NL
      NEWCPD=1
205   N=-1
      CALL COPYER(16,A,BUFFER,N)
      IF(N.EQ.0)GO TO 10
      K=NCTOT
      DO 210 I=1,N,2
      K=K+1
      CALL COPY8(PROPRT(K,1),A(I))
      CALL COPY8(PROPRT(K,2),A(I+1))
      J=NVNAME+I
      CALL COPY8(VNAME(J),A(I))
210   CALL COPY8(VNAME(J+1),A(I+1))
      NVNAME=NVNAME+N
      N=N/2
      NOCOMP=NOCOMP+N
      NCTOT=NCTOT+N
      NSLMAX=NSLMAX+N
      GO TO 10
C
C     NEW (NOT IN LIBRARY) COMPONENT NAMES
C
```

```
      NCTOT=NOCOMP
      NSLMAX=NL+NOCOMP
      NVNAME=NSLMAX+NSLMAX
      NEWCPD=1
225   N=-1
      CALL COPYER(16,A,BUFFER,N)
      IF(N.EQ.0)GO TO 10
      K=NCTOT
      DO 230 I=1,N,2
      K=K+1
      CALL COPY8(PROPRT(K,1),A(I))
      CALL COPY8(PROPRT(K,2),A(I+1))
      J=NVNAME+I
      CALL COPY8(VNAME(J),A(I))
      CALL COPY8(VNAME(J+1),A(I+1))
230   CONTINUE
      NVNAME=NVNAME+N
      N=N/2
      NSLMAX=NSLMAX+N
      NCMORE=NCMORE+N
      NCTOT=NCTOT+N
      GO TO 10
C
C     ADDITIONAL PROPERTIES
C
235   READ(NR,1000)BUFFER
237   N=2
      CALL COPYER(16,A,BUFFER,N)
      DO 240 I=1,NCTOT
      CALL COMP8(A(1),PROPRT(I,1),ICOMP)
      IF(ICOMP.NE.1)GO TO 240
      CALL COMP8(A(2),PROPRT(I,2),ICOMP)
      IF(ICOMP.EQ.1)GO TO 242
240   CONTINUE
      I=IC
      GO TO 243
242   NC=0
      READ(NR,1000)BUFFER
243   N=-1
      CALL COPYER(8,A,BUFFER,N)
```

```
      IF(N.EQ.0)GO TO 10
      READ(7,245)(USERPPT(I,J+NC),J=1,N)
245   FORMAT(30F0.0)
      NC=NC+N
      IF(NUSRPPT.LT.NC)NUSRPPT=NC
      IC=I
      GO TO 10
C
C     VARIABLE NAMES
C
250   READ(NR,1000)BUFFER
      NVNAME=-1
      CALL COPYER(16,VNAME,BUFFER,NVNAME)
      GO TO 10
255   N=-1
      CALL COPYER(16,A,BUFFER,N)
      DO 260  I=1,N,2
      J=I+NVNAME
      CALL COPY8(VNAME(J),A(I))
260   CALL COPY8(VNAME(J+1),A(I+1))
      NVNAME=NVNAME+N
      IVN=1
      GO TO 10
C
C     OPTION TO GO ON OR STOP IF A LOOP FAILS TO CONVERGE IN THE SPECIFIED
C     NUMBER OF LOOPS
C
C     GO ON  (KEEPON)  OPTION
280   NOGO=0
      GO TO 10
C     STOP RUN  (NOGO)  OPTION
285   NOGO=1
      GO TO 10
C
C     ROUTINE NAMES
C
300   READ(NR,1000)BUFFER
      N=2
      CALL COPYER(8,A,BUFFER,N)
310   DO 320 I=1,NEMAX
      CALL COMP8(A(1),ENAME(I),ICOMP)
      IF(ICOMP-1)320,340,320
```

```
 330  FORMAT(10X,22H ERROR IN ROUTINE NAME/10X,A8,32H IS NOT A UNIT NAME
     1 IN DATA CARD /1X,10A8///10X,22H SIMULATION ABANDONNED///)
      PAUSE ER
 340  CALL COPY8(RNAME(I),A(2))
      IPM=1
      IF(JSWTCH2.EQ.1)GO TO 10
      DO 342 J=1,21
      CALL COMP8(RNAME(I),SMODN(J),ICOMP)
      IF(ICOMP-1)342,346,342
 342  CONTINUE
      WRITE(NW,344)RNAME(I)
 344  FORMAT(//10X,23H ERROR IN MODULE NAME   //
     *10X,A8,29H  IS NOT A VALID MODULE NAME //)
      PAUSE ER
 346  MODULE(I)=J
      IRT=1
      GO TO 10
C
C     STREAMS
C
 350  READ(NR,1000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
 352  DO 354 IS=1,NSMAX
      CALL COMP8(A(1),SNAME(IS),ICOMP)
      IF(ICOMP-1)354,358,354
 354  CONTINUE
      WRITE(NWUSER,356)A(1),BUFFER
 356  FORMAT(///10X,22H ERROR IN STREAM ARRAY /
     110X,A8,37H IS NOT A STREAM NAME IN STREAM CARD /1X,10A8///
     230X,22H SIMULATION ABANDONNED///)
      PAUSE ER
 358  IS=(IS-1)*NSLMAX
      IF(N-1)382,382,359
 359  DO 360 I=1,8
      J=1
      CALL COMP(J,A(1),I,VNAMES(12),1)
      IF(J)360,360,362
 360  CONTINUE
 362  K=I/2
```

```
      GO TO (364,366,368,370),K
364   READ(7,365)(SMATRX(IS+L),L=1,N-1)
365   FORMAT(2X,30F0.0)
      GO TO 380
366   READ(7,367)(SMATRX(IS+L),L=1,N-1)
367   FORMAT(4X,30F0.0)
      GO TO 380
368   READ(7,369)(SMATRX(IS+L),L=1,N-1)
369   FORMAT(6X,30F0.0)
      GO TO 380
370   READ(7,371)(SMATRX(IS+L),L=1,N-1)
371   FORMAT(8X,30F0.0)
380   IF(N-NSLMAX)385,385,395
385   READ(NR,390)(SMATRX(IS+L),L=N,NSLMAX)
390   FORMAT(30F0,0)
395   GO TO 10
382   DO 384 I=1,NSLMAX
384   SMATRX(IS+I)=0.0
      GO TO 10
C
C     PREFERRED STREAMS
C
450   READ(NR,1000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
455   DO 490 I=1,N
      DO 460 J=1,NSMAX
      CALL COMP8(A(I),SNAME(J),ICOMP)
      IF(ICOMP-1)460,480,460
460   CONTINUE
      WRITE(NWUSER,470)A(I),BUFFER
470   FORMAT(10X,31H ERROR IN PREFERRED STREAM NAME /10X,A8,35H IS NOT A
     1STREAM NAME, IN DATA CARD  /1X,10A8,///10X,22H SIMULATION ABANDONN
     2ED///)
      PAUSE ER
480   KPS(NPS+I)=J
      KSFLAG(J)=2
490   CONTINUE
      NPS=NPS+N
      GO TO 10
C
C     THE CALCULATION ORDER
```

```
500     READ(NR,1000)BUFFER
        N=-1
        CALL COPYER(8,A,BUFFER,N)
501     I=4
        CALL COMP(I,A(1),1,VNAMES(1),1)
        IF(I-4)505,502,505
502     ISEQ=2
        ISQ=1
        GO TO 510
505     I=3
        CALL COMP(I,A(1),1,VNAMES(2),1)
        IF(I-3)510,508,510
508     ISEQ=1
        ISQ=1
        IF(NNE.LT.1)GO TO 510
509     NELIST(NNE-1,2)=3
510     DO 525 J=1,N
        IF(ISQ-1)512,511,512
511     ISQ=0
        GO TO 525
512     DO 515 K=1,NEMAX
        CALL COMP8(A(J),ENAME(K),ICOMP)
        IF(ICOMP-1)515,520,515
515     CONTINUE
        WRITE(NWUSER,518)A(J),BUFFER
518     FORMAT(///10X,32H ERROR IN CALCULATION ORDER LIST/
       110X,A8,34H IS NOT A UNIT NAME IN DATA CARD  /1X,10A8///
       230X,22H SIMULATION ABANDONNED///)
        PAUSE ER
520     NNE=NNE+1
        NELIST(NNE,1)=K
        NELIST(NNE,2)=ISEQ
525     CONTINUE
        NLIST=NNE
        IF(NLIST-NEMAX)500,530,530
530     IF(NELIST(NLIST,2).EQ.2)NELIST(NLIST,2)=3
        GO TO 10
C
C       NO CALCULATION ORDER
C
550     NLIST=0
```

```
      GO TO 10
C
C     CALCULATE
C
600   IERR=N-2
      IF(NEWCPD)900,900,605
605   NC=0
      KALL=0
      NRF=NRCOMP
      IF(NOCOMP.EQ.0)GO TO 644
610   READ(NRF,910)NCOMP
      READ(NRF,910)NPROP
      READ(NRCOMPX,910)NCOMPX
      READ(NRCOMPX,910)NPROPX
      DO 630 I=1,NCOMP
      READ(NRF,1000)A(1),A(2)
      READ(NRCOMPX,1000)A(3)
      DO 613 J=1,NCTOT
      CALL COMP8(A(1),PROPRT(J,1),ICOMP)
      IF(ICOMP.NE.1)GO TO 613
612   CALL COMP8(A(2),PROPRT(J,2),ICOMP)
      IF(ICOMP.EQ.1)GO TO 614
613   CONTINUE
      GO TO 624
614   NC=NC+1
      READ(NRF,615)(PROPRT(J,M),M=3,NPROP)
      READ(NRCOMPX,615)(PRTEXTRA(J,M),M=1,NPROPX)
615   FORMAT(1E15.7)
      GO TO 629
624   DO 625 I2=3,NPROP
625   READ(NRF,615)ZZ
      DO 626 I2=1,NPROPX
626   READ(NRCOMPX,615)ZZ
629   IF(NC.EQ.NCTOT)GO TO 645
630   CONTINUE
635   IF(NC.GE.NCTOT)GO TO 645
      IF(NC.GE.NOCOMP)GO TO 644
      IF(KALL.GT.0)GO TO 637
      KALL=1
      NRF=NRCOMP2
      PAUSE FILEC
      GO TO 610
```

```
640   FORMAT([illegible]
     1AMES,3(/10A8)///30X,22H SIMULATION ABANDONNED///)
642   PAUSE FR
644   IF(KALL.EQ.2)GO TO 637
      IF(NCTOT.EQ.NOCOMP)GO TO 645
      KALL=2
      PAUSE FILEB
      NRF=NRCOMP1
      NRCOMPX=NRCOMPX1
      GO TO 610
645   IF(IVN.EQ.1)GO TO 646
      I3=IUNITS*3
      NRCOMPX=NRCOMPX2
      CALL COPY8(VNAME(4),VNAME2(I3+1))
      CALL COPY8(VNAME(6),VNAME2(I3+2))
      CALL COPY8(VNAME(8),VNAME2(I3+3))
646   IF(IUNITS.EQ.0)GO TO 900
      DO 647 I=1,NOCOMP
      PROPRT(I,5)=PROPRT(I,5)/14.5
      PROPRT(I,6)=PROPRT(I,6)/1.8
      PROPRT(I,7)=PROPRT(I,7)/16.03
      PROPRT(I,8)=PROPRT(I,8)/1.8
      PRTEXTRA(I,1)=PRTEXTRA(I,1)/1.8
      IF(PRTEXTRA(I,1).EQ.0.0)PRTEXTRA(I,3)=(PRTEXTRA(I,3)-32.0)/1.8
      PRTEXTRA(I,8)=PRTEXTRA(I,8)/1.8
      PRTEXTRA(I,29)=PRTEXTRA(I,29)/1.8
      DO 647 J=14,19
      PROPRT(I,J)=PROPRT(I,J)*2.325
647   CONTINUE
      GO TO 900
C
C     STREAM LENGTH
C
650   N=-4
      CALL COPYER(8,A,BUFFER,N)
      NSLMAX=A(4)
      GO TO 10
C
C     TYPE OF UNITS USED
C
C     BRITISH UNITS
```

```
670   IUNITS=0
      GO TO 10
C     S.I. UNITS
680   IUNITS=1
      GO TO 10
C
C     CONVERGENCE PROMOTER
C
700   IF(N.GT.2)GO TO 705
      KONVRG=1
      GO TO 10
705   N=-3
      CALL COPYER(8,A,BUFFER,N)
      KONVRG=A(3)
      GO TO 10
C
C     NO CONVERGENCE PROMOTION
C
710   KONVRG=0
      GO TO 10
C
C     TOLERANCE
C
720   N=-3
      CALL COPYER(8,A,BUFFER,N)
      TOL=A(3)/100.
      GO TO 10
C
C     NUMBER OF LOOPS
C
750   N=-2
      CALL COPYER(8,A,BUFFER,N)
      NLOOPS=A(2)
      GO TO 10
C
C     TRACE
C
770   IF(N-1)772,772,774
772   ITRACE=0
      GO TO 10
774   N=-2
      CALL COPYER(8,A,BUFFER,N)
```

```
      GO TO 10
C
C     NO TRACE
C
790   ITRACE=-1
      GO TO 10
C
C     DATA ECHO
C
800   WRITE(NWUSER,802)TITLE
802   FORMAT(1H1//1X,10A8/1X,80(1H*)/)
      GO TO 900
803   WRITE(NWUSER,804)
804   FORMAT(//10X,15H PROCESS MATRIX/11X,14(1H*)/)
      DO 820 I=1,NEMAX
      IF(KPM(I,1).EQ.0)GO TO 820
      WRITE(NWUSER,806)ENAME(I)
806   FORMAT(/10X,13H UNIT NAME : ,A8)
      DO 812 J=2,NCOLPM
      JK=KPM(I,J)
      IF(JK)814,814,808
808   J1=J-1
      WRITE(NWUSER,810)J1,SNAME(JK)
810   FORMAT(15X,18H INPUT  STREAM NO.,I2,5X,A8)
812   CONTINUE
814   DO 819 K=J,NCOLPM
      JK=IABS(KPM(I,K))
      IF(JK)820,820,816
816   K1=K-J+1
      WRITE(NWUSER,818)K1,SNAME(JK)
818   FORMAT(15X,18H OUTPUT STREAM NO. ,I2,5X,A8)
819   CONTINUE
820   CONTINUE
817   WRITE(NWUSER,821)
821   FORMAT(//10X,14H PLANT FEEDS : /11X,13(1H*)/)
      DO 822 I=1,NSMAX
      IF(KSEM(I,2).EQ.0.AND.KSEM(I,3).NE.0)
     *WRITE(NWUSER,827)SNAME(I)
822   CONTINUE
      WRITE(NWUSER,823)
823   FORMAT(//10X,16H PLANT OUTPUTS : /11X,15(1H*)/)
```

```
      DO 825 I=1,NSMAX
      IF(KSEM(I,3).EQ.0.AND.KSEM(I,2).NE.0)
     *WRITE(NWUSER,827)SNAME(I)
  825 CONTINUE
  827 FORMAT(16X,A8)
      WRITE(NWUSER,824)
  824 FORMAT(//10X,14H STREAM MATRIX/11X,13(1H*))
      NLM=NSMAX/3
      NM=NSMAX-NLM*3
      IF(NLM)836,836,826
  826 I1=-2
      DO 834 I=1,NLM
      I1=I1+3
      I2=I1+2
      WRITE(NWUSER,828)(SNAME(K),K=I1,I2)
  828 FORMAT(//32X,3(7X,A8)/)
      J1=-1
      DO 832 J=1,NSLMAX
      J1=J1+2
      WRITE(NWUSER,830)VNAME(J1),VNAME(J1+1),(SMATRX((L-1)*NSLMAX+J)
     *,L=I1,I2)
  830 FORMAT(11X,2A8,4X,3F15.5)
  832 CONTINUE
  834 CONTINUE
  836 IF(NM.EQ.0)GO TO 839
      NLM=NLM*3+1
      WRITE(NWUSER,828)(SNAME(K),K=NLM,NSMAX)
      WRITE(NWUSER,837)
  837 FORMAT(10X)
      J1=-1
      DO 838 J=1,NSLMAX
      J1=J1+2
  838 WRITE(NWUSER,830)VNAME(J1),VNAME(J1+1),(SMATRX((L-1)*NSLMAX+J)
     *,L=NLM,NSMAX)
  839 WRITE(NWUSER,840)
  840 FORMAT(//10X,17H UNITS PARAMETERS/11X,16(1H*)/)
      IF(NPAR.GT.0)GO TO 842
      WRITE(NWUSER,841)
  841 FORMAT(/10X,50H ** NO PARAMETERS HAVE BEEN GIVEN FOR THE UNITS **
     */)
      GO TO 847
  842 DO 844 I=1,NEMAX
```

```
      IF(K.EQ.0)GO TO 843
      K=K+J-1
      WRITE(NWUSER,845)ENAME(I),(EEN(L),L=J,K)
      GO TO 844
843   WRITE(NWUSER,849)ENAME(I)
844   CONTINUE
845   FORMAT(/10X,2X,A8,2X,4F15.5,7(/20X,4F15.5))
849   FORMAT(/10X,2X,A8,5X,'HAS NO PARAMETERS')
847   WRITE(NWUSER,846)
846   FORMAT(//10X,23H UNITS AND MODELS NAMES/11X,22(1H*)/)
      IF(IRT.GT.0)GO TO 848
      WRITE(NWUSER,8510)
8510  FORMAT(/10X,40H ** NO ROUTINES NAMES HAVE BEEN GIVEN ** /)
      GO TO 853
848   DO 851 I=1,NEMAX
      CALL COMP8(RNAME(I),ZERO,ICOMP)
      IF(ICOMP.EQ.1)GO TO 850
      WRITE(NWUSER,852)ENAME(I),RNAME(I)
      GO TO 851
850   WRITE(NWUSER,855)ENAME(I)
855   FORMAT(10X,' UNIT ' ,2X,A8,2X,' HAS NO MODEL ASSIGNED TO IT')
851   CONTINUE
852   FORMAT(10X,8H UNIT   ,A8,28H   REPRESENTED BY ROUTINE   ,A8)
853   WRITE(NWUSER,854)
854   FORMAT(//10X,19H PROBLEM DIMENSIONS/11X,18(1H*)/)
      WRITE(NWUSER,856)NEMAX,NSMAX,NOCOMP,NSLMAX
856   FORMAT(/10X,18H NUMBER OF UNITS = ,I8
     1/10X,20H NUMBER OF STREAMS = ,I6
     2/10X,24H NUMBER OF COMPONENTS = ,I2
     3/10X,17H STREAM LENGTH = ,I9)
      WRITE(NWUSER,858)
858   FORMAT(//10X,17H PROBLEM CONTROLS/11X,16(1H*)/)
      WRITE(NWUSER,860)NLOOPS,TOL
860   FORMAT(10X,34H MAXIMUM NUMBER OF LOOPS ALLOWED = ,I5/
     110X,20H TOLERANCE ALLOWED = ,F10.5)
      IF(KONVRG)862,862,866
862   WRITE(NWUSER,864)
864   FORMAT(10X,29H CONVERGENCE PROMOTER NOT SET)
      GO TO 870
866   WRITE(NWUSER,868)KONVRG
```

```
  868 FORMAT(10X,25H CONVERGENCE PROMOTER NO.,I2,4H SET)
  870 IF(ITRACE)876,872,873
  872 WRITE(NWUSER,874)
      GO TO 880
  873 WRITE(NWUSER,875)ITRACE
      GO TO 880
  874 FORMAT(10X,23H TRACE SET ON ALL UNITS)
  875 FORMAT(10X,17H TRACE SET EVERY  ,I3,6H LOOPS  )
  876 WRITE(NWUSER,878)
  878 FORMAT(10X,21H TRACE OPTION NOT SET)
  880 IF(NLIST)882,882,886
  882 WRITE(NWUSER,884)
  884 FORMAT(//10X,28H CALCULATION ORDER NOT GIVEN/11X,27(1H*)/)
      GO TO 898
  886 WRITE(NWUSER,888)
  888 FORMAT(//10X,18H CALCULATION ORDER/11X,17(1H*))
      ISEQ=2-NELIST(1,2)
      DO 896 I=1,NLIST
      IS2=NELIST(I,2)/2
      IF(IS2-ISEQ)890,894,892
  890 WRITE(NWUSER,891)
  891 FORMAT(/12X,15H SEQUENTIAL SET/)
      ISEQ=0
      GO TO 894
  892 WRITE(NWUSER,893)
  893 FORMAT(/12X,13H RECYCLE LOOP/)
      ISEQ=1
  894 K1=NELIST(I,1)
      WRITE(NWUSER,895)ENAME(K1)
  895 FORMAT(11X,A8)
  896 CONTINUE
  898 WRITE(NWUSER,899)
  899 FORMAT(///30X,17H END OF DATA ECHO  /31X,16(1H*)///)
      GO TO 10
C
C     SEND DATA TO LOG FILE
C
  900 DO 901 I=1,NEMAX
      DO 901 J=1,3
  901 KSEM(I,J)=0
      DO 907 I=1,NEMAX
      IF(KPM(I,1))907,907,902
```

```
      DO 906 K=2,NOUPM
      IF(KPM(I,K))904,907,903
903   I=KPM(I,K)
      KSEM(L,3)=KPM(I,1)
      GO TO 905
904   I=-KPM(I,K)
      KSEM(L,2)=KPM(I,1)
905   KSEM(L,1)=L
906   CONTINUE
907   CONTINUE
      IF(NLIST.EQ.0)CALL ORDER(NW)
      IF(INDEX.EQ.27)GO TO 803
      IF(JSWTCH2.EQ.2)GO TO 908
      PAUSE EQCAL
      CALL EQCALLER(NWFILE)
908   PAUSE FILEA
      GO TO 909
909   IF(NPAR.LT.1)WRITE(NWUSER,3000)
      IF(IRT.EQ.0)WRITE(NWUSER,3010)
      IF(NCTOT.EQ.0)WRITE(NWUSER,3020)
      CALL FILER(NWFILE,INDEX)
      STOP
C
C
C *** ERRORS IN STREAM MEASUREMENTS FOR DATA RECONCILIATION
C
1001  READ(NR,1000)BUFFER
1005  N=-2
      CALL COPYER(8,A,BUFFER,N)
      DO 1010 I=1,NSMAX
      CALL COMP8(A(1),SNAME(I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1030
1010  CONTINUE
      WRITE(NW,1020)A(1)
1020  FORMAT(//10X,'ERROR IN ERRORS DATA'/
     *10X,A8,'IS NOT A RECOGNISED STREAM NAME'//
     *30X,'SIMULATION ABANDONNED'//)
      PAUSE ER
1030  ERR(I)=A(2)/100.
      GO TO 10
C
```

```
C
C     INITIAL READ SEGMENT
C
910   FORMAT(16I5)
930   FORMAT(5E15.7)
C
2000  READ(NRFILE,1000)TITLE
      READ(NRFILE,910)NEMAX,NSMAX,NOCOMP,NSLMAX,NPAR,NCOLPM,KONVRG,
     1NPS,ITRACE,NLOOPS,NLIST,NVNAME,NPROP,NOGO,IUNITS,NUSRPPT,NCTOT
     2,NPROPX,IRT,IERR
      READ(NRFILE,930)TOL
      READ(NRFILE,1000)(SNAME(I),I=1,NSMAX)
      READ(NRFILE,1000)(ENAME(I),I=1,NEMAX)
      IF(IRT.EQ.0)GO TO 2005
      READ(NRFILE,1000)(RNAME(I),I=1,NEMAX)
      READ(NRFILE,910)(MODULE(I),I=1,NEMAX)
2005  READ(NRFILE,1000)(VNAME(I),I=1,NVNAME)
      IF(NOCOMP)2020,2020,2010
2010  DO 2011 I=1,NOCOMP
      READ(NRFILE,1000)(PROPRT(I,J),J=1,2)
      READ(NRFILE,930)(PROPRT(I,J),J=3,NPROP)
2011  READ(NRFILE,930)(PRTEXTRA(I,J),J=1,NPROPX)
      IF(NUSRPPT)2020,2020,2012
2012  DO 2013 I=1,NCTOT
2013  READ(NRFILE,930)(USERPPT(I,J),J=1,NUSRPPT)
2020  READ(NRFILE,910)((KPM(I,J),J=1,NCOLPM),I=1,NEMAX)
      READ(NRFILE,910)((KSEM(I,J),J=1,3),I=1,NSMAX)
      READ(NRFILE,930)(SMATRX(I),I=1,NSLMAX*NSMAX)
      IF(NPAR.LT.1)GO TO 2025
      READ(NRFILE,930)(EEN(I),I=1,NPAR)
      READ(NRFILE,910)((NEMAT(I,J),J=1,2),I=1,NEMAX)
2025  IF(NLIST)2040,2040,2030
2030  READ(NRFILE,910)((NELIST(I,J),J=1,2),I=1,NLIST)
      READ(NRFILE,910)(KSFLAG(I),I=1,NSMAX)
2040  IF(NPS)2060,2060,2050
2050  READ(NRFILE,910)(KPS(I),I=1,NPS)
2060  IF(IERR.EQ.0)GO TO 10
      READ(NRFILE,930)(ERR(I),I=1,NSMAX)
      GO TO 10
C
C
C
```

```
C
 2100 READ(NR,1000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
 2101 NEMAX=NEMAX+1
      NU=NEMAX
      CALL COPY8(ENAME(NEMAX),A(1))
      DO 2110 I=2,N
      IF(NSMAX)2106,2106,2102
 2102 DO 2104 J=1,NSMAX
      CALL COMP8(SNAME(J),A(I),ICOMP)
      IF(ICOMP-1)2104,2108,2104
 2104 CONTINUE
 2106 NSMAX=NSMAX+1
      CALL COPY8(SNAME(NSMAX),A(I))
      J=NSMAX
 2108 NCOLMX=NCOLMX+1
      KPM(NU,I)=J
 2110 CONTINUE
      KPM(NU,1)=NU
      READ(NR,1000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
      IF(N.EQ.0)GO TO 2118
      DO 2116 I=1,N
      DO 2112 J=1,NSMAX
      CALL COMP8(SNAME(J),A(I),ICOMP)
      IF(ICOMP-1)2112,2114,2112
 2112 CONTINUE
      NSMAX=NSMAX+1
      CALL COPY8(SNAME(NSMAX),A(I))
      J=NSMAX
 2114 NCOLMX=NCOLMX+1
      KPM(NU,NCOLMX)=-J
 2116 CONTINUE
 2118 IF(NCOLPM-NCOLMX)2120,2130,2130
 2120 NCOLPM=NCOLMX
 2130 NCOLMX=1
      GO TO 10
C
C     EQUIPMENT PARAMETERS FOR THE READING SECTION
```

```
C
2150  READ(NR,1000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
2152  DO 2154 IE=1,NEMAX
      CALL COMP8(A(1),ENAME(IE),ICOMP)
      IF(ICOMP-1)2154,2158,2154
2154  CONTINUE
2156  N1=N
      IE=NIE
      GO TO 2166
2158  IF(N.LT.2)GO TO 10
      DO 2160 I=1,8
      J=1
      CALL COMP(J,A(1),I,VNAMES(12),1)
      IF(J)2160,2160,2162
2160  CONTINUE
2162  K=I/2
      N1=N-1
      NP1=NPAR+1
      GO TO (2172,2176,2180,2184),K
2172  READ(7,2174)(EEN(NPAR+I),I=1,N1)
2174  FORMAT(2X,30F0.0)
      GO TO 2170
2176  READ(7,2178)(EEN(NPAR+I),I=1,N1)
2178  FORMAT(4X,30F0.0)
      GO TO 2170
2180  READ(7,2182)(EEN(NPAR+I),I=1,N1)
2182  FORMAT(6X,30F0.0)
      GO TO 2170
2184  READ(7,2186)(EEN(NPAR+I),I=1,N1)
2186  FORMAT(8X,30F0.0)
      GO TO 2170
2166  READ(7,2168)(EEN(NPAR+I),I=1,N1)
2168  FORMAT(100F0.0)
2170  NPAR=NPAR+N1
      NEMAT(IE,1)=NP1
      NEMAT(IE,2)=NEMAT(IE,2)+N1
      NIE=IE
      GO TO 10
C
3000  FORMAT(//10X,16H *** WARNING ***  /
```

```
3010  FORMAT(//10X,...
     *10X,33H *** MODELS NAMES NOT SET YET ***    //)
3020  FORMAT(//10X,16H *** WARNING ***  /
     *10X,34H *** COMPONENT NAMES NOT GIVEN ***    //)
      END
      SUBROUTINE EQCALLER(NWF)
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
      COMMON /CNAME/ SNAME(30),ENAME(30),RNAME(30),VNAME(60),
     1MODULE(30),TITLE(10),NSMAX,NEMAX,NPAR,NVNAME,ITRACE,NPROP,NUSRPPT
      PAUSE EQCAL
      WRITE(NWF,10)
10    FORMAT(6X,21HSUBROUTINE EQCALL(NE)/
     *6X,7HGO TO ()
      DO 20 I=1,NEMAX-1
20    WRITE(NWF,30)I
30    FORMAT(5X,1H*,I2,1H,)
      WRITE(NWF,40)
40    FORMAT(5X,1H*,I2,4H),NE)
      DO 50 I=1,NEMAX
50    WRITE(NWF,60)I,RNAME(I)
60    FORMAT(1X,I2,3X,5HCALL ,A8/6X,6HRETURN)
      WRITE(NWF,70)
70    FORMAT(6X,3HEND)
      RETURN
      END
      SUBROUTINE FILER(NWFILE,INDEX)
C
C
      COMMON /DIMS1/
     *TITLE(10),BUFFER(10),VNAMES(31),A(20),
     *EFN(500),NEMAT(30,2),SMATRX(400),RNAME(30),VNAME(30),MODULE(30),
     *PROPRT(10,25),PRTEXTRA(10,35),USERPPT(10,10),SMODN(20),FORMS(2)
     *.VNAME2(6),ERR(30),ZERO
C
      COMMON /DIMS2/
     * NOCOMP,NSLMAX,NPAR,KONVRG,NPROPX
     *.ITRACE,NLOOPS,NVNAME,NPROP,NOGO,IUNITS
     *.NUSRPPT,NCTOT,IRT,IERR
C
      COMMON
```

```
     *KPM(30,8),KSFLAG(30),KSEM(30,3),NELIST(30,2),KPS(10),
     *NEMAX,NSMAX,NCOLPM,NLIST,NPS,ENAME(30),SNAME(30),TOL
C
C
      WRITE(NWFILE,1000)TITLE
      WRITE(NWFILE,910)NEMAX,NSMAX,NOCOMP,NSLMAX,NPAR,NCOLPM,KONVRG,
     1NPS,ITRACE,NLOOPS,NLIST,NVNAME,NPROP,NOGO,IUNITS,NUSRPPT,NCTOT
     *,NPROPX,IRT,IERR
910   FORMAT(16I5)
      WRITE(NWFILE,930)TOL
      WRITE(NWFILE,1000)(SNAME(I),I=1,NSMAX)
      WRITE(NWFILE,1000)(ENAME(I),I=1,NEMAX)
      IF(IRT.EQ.0)GO TO 915
      WRITE(NWFILE,1000)(RNAME(I),I=1,NEMAX)
      WRITE(NWFILE,910)(MODULE(I),I=1,NEMAX)
915   WRITE(NWFILE,1000)(VNAME(I),I=1,NVNAME)
      IF(NOCOMP)935,935,925
925   DO 926 I=1,NOCOMP
      WRITE(NWFILE,1000)(PROPRT(I,J),J=1,2)
      WRITE(NWFILE,930)(PROPRT(I,J),J=3,NPROP)
926   WRITE(NWFILE,930)(PRTEXTRA(I,J),J=1,NPROPX)
930   FORMAT(5F15.7)
      IF(NUSRPPT)935,935,931
931   DO 932 I=1,NCTOT
932   WRITE(NWFILE,930)(USERPPT(I,J),J=1,NUSRPPT)
935   WRITE(NWFILE,910)((KPM(I,J),J=1,NCOLPM),I=1,NEMAX)
      WRITE(NWFILE,910)((KSEM(I,J),J=1,3),I=1,NSMAX)
      WRITE(NWFILE,930)(SMATRX(I),I=1,NSLMAX*NSMAX)
      IF(NPAR.LT.1)GO TO 938
      WRITE(NWFILE,930)(EEN(I),I=1,NPAR)
      WRITE(NWFILE,910)((NEMAT(I,J),J=1,2),I=1,NEMAX)
938   IF(NLIST)950,950,940
940   WRITE(NWFILE,910)((NELIST(I,J),J=1,2),I=1,NLIST)
      WRITE(NWFILE,910)(KSFLAG(I),I=1,NSMAX)
950   IF(NPS)960,960,955
955   WRITE(NWFILE,910)(KPS(I),I=1,NPS)
960   IF(IERR.EQ.0)GO TO 965
      WRITE(NWFILE,930)(ERR(I),I=1,NSMAX)
965   IF(INDEX.EQ.0)PAUSE ER
      IF(INDEX.EQ.28)PAUSE EXIT
1000  FORMAT(10A8)
      PAUSE
```

```
C
      DIMENSION LOOP(20,30),IP(30),ISTBUF(30)
      COMMON
     *KPM(30,8),KSFLAG(30),KSEM(30,3),NELIST(30,2),KPS(10),
     *NEMAX,NSMAX,NCOLPM,NLIST,NPS,ENAME(30),SNAME(30)
      COMMON /ORDR/ IH(30,30),KFLAG1,LOOPN,NE1,IE,KEFLAG(30)
C
      DO 5 K=1,NSMAX
      KSFLAG(K)=0
      I=KSEM(K,2)
      J=KSEM(K,3)
5     ISTBUF(K)=I*100+J
      DO 20 I=1,NEMAX
      IP(I)=0
      DO 10 J=1,20
10    LOOP(J,I)=0
      DO 20 J=1,NEMAX
      IH(I,J)=0
20    CONTINUE
C
C     ***  EC 1  ***
C
      NLCUML=0
      NLOOPS=0
      K=1
      IP(1)=1
C
C     ***  EC 2  ***
C
30    N=IP(K)
      DO 50 J=2,NCOLPM
      L=-KPM(N,J)
      IF(L.EQ.0)GO TO 70
      IF(L.LT.0)GO TO 50
      L=KSEM(L,3)
      IF(L.LE.IP(1))GO TO 50
      DO 40 M=1,NEMAX
      IF(L.EQ.IP(M))GO TO 50
      I=IP(K)
      IF(L.EQ.IH(I,M))GO TO 50
```

```
40      CONTINUE
        GO TO 60
50      CONTINUE
        GO TO 70
60      K=K+1
        IP(K)=L
        GO TO 30
C
C       ***  EC 3  ***
C
70      N=IP(K)
        DO 80 J=2,NCOLPM
        L=-KPM(N,J)
        IF(L.EQ.0)GO TO 110
        IF(L.LT.0)GO TO 80
        L=KSEM(L,3)
        IF(IP(1).EQ.L)GO TO 90
80      CONTINUE
        GO TO 110
C
90      NLOOPS=NLOOPS+1
        DO 100 J=1,NEMAX
100     LOOP(NLOOPS,J)=IP(J)
C
C       ***  EC 4  ***
C
110     IF(K.EQ.1)GO TO 150
        L=IP(K)
        DO 120 M=1,NEMAX
120     IH(L,M)=0
        L=IP(K-1)
        DO 130 M=1,NEMAX
        IF(IH(L,M).EQ.0)GO TO 140
130     CONTINUE
C
140     IH(L,M )=IP(K)
        IP(K)=0
        K=K-1
        GO TO 30
C
C       ***  EC 5  ***
C
```

```
      K=1
      DO 160 I=1,NEMAX
      DO 160 J=1,NEMAX
160   IH(I,J)=0
      GO TO 30
C
180   IF(NLOOPS.LT.1)GO TO 1500
      DO 190 I=1,NEMAX
      DO 190 J=1,NEMAX
190   IH(I,J)=0
200   DO 270 L=1,NLOOPS
      DO 220  I=1,NEMAX
      IF(LOOP(L,I).GT.0)GO TO 210
      IP(I)=IP(1)
      GO TO 230
210   IP(I)=LOOP(L,I)
      LOOP(L,I)=0
220   CONTINUE
230   I=I-1
      DO 260 J=1,I
      J1=IP(J)
      IH(L,J1)=1
      J2=IP(J+1)
      IST=J1*100+J2
      IP(J)=0
      DO 240 K=1,NSMAX
      IF(ISTBUF(K).EQ.IST)GO TO 250
240   CONTINUE
250   LOOP(L,K)=1
260   CONTINUE
      IP(I+1)=0
270   CONTINUE
C
C *** CUMULATIVE MATRIX
C
      NLCUML=NLOOPS
      N=0
      DO 450 I=1,NLCUML
400   DO 440 J=1,NEMAX
      IF(IH(I,J).LT.1)GO TO 440
```

```
      IF(I.GE.NLCUML)GO TO 460
      DO 430 K=I+1,NLCUML
      IF(IH(K,J).LT.1)GO TO 430
      DO 420 L=1,NEMAX
      IH(I,L)=IH(I,L)+IH(K,L)
      IH(K,L)=0
      IF(K.EQ.NLCUML)GO TO 420
      DO 410 M=K+1,NLCUML
410   IH(M-1,L)=IH(M,L)
420   CONTINUE
425   N=1
      NLCUML=NLCUML-1
430   CONTINUE
440   CONTINUE
      IF(N.EQ.0)GO TO 450
      N=0
      GO TO 400
450   CONTINUE
460   DO 500 I=1,NLCUML
      IH(I,NEMAX+1)=0
      DO 500 J=1,NEMAX
      IF(IH(I,J).GT.0)IH(I,NEMAX+1)=IH(I,NEMAX+1)+1
500   CONTINUE
C
C *** SEARCH FOR ITERATE STREAMS TO CUT ALL LOOPS
C
      NS=0
      N1=-1
C *** GET NUMBER OF LOOPS IN WHICH EACH STREAM IS INVOLVED
      DO 1100 I=1,NSMAX
      DO 1100 L=1,NLOOPS
1100  LOOP(NLOOPS+1,I)=LOOP(NLOOPS+1,I)+LOOP(L,I)
C *** FLAG PREFERRED STREAMS  100  IN ROW NLOOPS+1 OF ITERATE STREAM SEARCH
C *** MATRIX
      IF(NPS)1130,1130,1110
1110  DO 1120 I=1,NPS
      J=KPS(I)
      LOOP(NLOOPS+1,J)=100
      KSFLAG(J)=2
1120  NS=NS+1
1130  N1=N1+1
      DO 1230 I=1,NSMAX
```

```
1140  NS=NS+1
      IF(N1)1230,1230,1150
1150  DO 1200 J=1,NLOOPS
      IF(LOOP(J,I))1200,1200,1160
1160  DO 1190 K=1,NSMAX
      IF(K-I)1170,1190,1170
1170  IF(LOOP(J,K).NE.1)GO TO 1190
1180  LOOP(J,I)=2
      JJ=JJ+1
      GO TO 1200
1190  CONTINUE
      GO TO 1220
1200  CONTINUE
      IF(JJ-N1)1220,1210,1210
1210  LOOP(NLOOPS+1,I)=0
      GO TO 1230
1220  LOOP(NLOOPS+1,I)=100
      NPS=NPS+1
      KPS(NPS)=I
      DO 1225 J=1,NLOOPS
      IF(LOOP(J,I).EQ.2)LOOP(J,I)=1
1225   CONTINUE
      KSFLAG(I)=2
1230  CONTINUE
      IF(NS.LT.NSMAX)GO TO 1130
      DO 1480 I=1,NSMAX
      IF(KSEM(I,2).GT.0)GO TO 1480
      KSFLAG(I)=1
1480  CONTINUE
C
C *** GET CALCULATION ORDER
C
1500  KFLAG=0
      DO 1600 IE=1,NEMAX
      IF(KEFLAG(IE).GT.0)GO TO 1600
      DO 1510 I=2,NCOLPM
      J=KPM(IE,I)
      IF(J.LE.0)GO TO 1520
      IF(KSFLAG(J).EQ.0)GO TO 1600
1510   CONTINUE
```

```
1520  IF(NLCUML.LT.1)GO TO 1570
      DO 1550 LOOPN=1,NLCUML
      IF(IH(LOOPN,IE).GT.0)GO TO 1560
1550  CONTINUE
      GO TO 1570
1560  KFLAG1=0
      CALL FLAGER
      IF(KFLAG1.EQ.1)KFLAG=1
      GO TO 1600
1570  DO 1580 I1=I,NCOLPM
      J=-KPM(IE,I1)
      IF(J.EQ.0)GO TO 1590
1580  KSFLAG(J)=1
1590  KFLAG=1
      NE1=NE1+1
      NELIST(NE1,1)=IE
      NELIST(NE1,2)=1
      KEFLAG(IE)=1
1600  CONTINUE
      IF(KFLAG.EQ.1)GO TO 1500
      IF(NE1.LT.NEMAX)GO TO 1500
      NLIST=NEMAX
C     WRITE(2,1700)((NELIST(I,J),J=1,2),I=1,NEMAX)
1700  FORMAT(10X,6H ORDER,12I5)
      RETURN
      END
      SUBROUTINE FLAGER
C
      COMMON
     *KPM(30,8),KSFLAG(30),KSEM(30,3),NELIST(30,2),KPS(10),
     *NEMAX,NSMAX,NCOLPM,NLIST,NPS,ENAME(30),SNAME(30)
      COMMON /ORDR/ IH(30,30),KFLAG1,LOOPN,NE1,IE,KEFLAG(30)
C
      NEFL=0
      KFLAG=0
      NEL=IH(LOOPN,NEMAX+1)
      K1=IE-1
20    DO 200 I=1,NEMAX
30    K1=K1+1
      IF(K1.GT.NEMAX)K1=1
      IF(IH(LOOPN,K1).LE.0)GO TO 30
40    IF(KEFLAG(K1).GT.0)GO TO 200
```

```
      IF(K3.LE.0)GO TO 60
      IF(KSFLAG(K3).LE.0)GO TO 200
50    CONTINUE
60    KEFLAG(K1)=3
      NE1=NE1+1
      NELIST(NE1,1)=K1
      NELIST(NE1,2)=2
      NEFL=NEFL+1
      KFLAG=1
      DO 70 J1=J,NCOLPM
      K3=-KPM(K1,J1)
      IF(K3.EQ.0)GO TO 200
      IF(KSFLAG(K3).EQ.0)KSFLAG(K3)=3
70    CONTINUE
200   CONTINUE
      IF(KFLAG.EQ.0)GO TO 210
      KFLAG=0
      K1=0
      GO TO 20
210   IF(NEFL.NE.NEL)GO TO 250
      KFLAG1=1
      DO 220 I=1,NSMAX
      IF(KSFLAG(I).EQ.3)KSFLAG(I)=1
220   CONTINUE
      DO 230 I=1,NEMAX
      IF(KEFLAG(I).EQ.3)KEFLAG(I)=1
230   CONTINUE
      NELIST(NE1,2)=3
      RETURN
250   DO 260 I=1,NEMAX
      IF(KEFLAG(I).EQ.3)KEFLAG(I)=0
260   CONTINUE
      DO 270 I=1,NSMAX
      IF(KSFLAG(I).EQ.3)KSFLAG(I)=0
270   CONTINUE
      KFLAG1=0
      NE1=NE1-NEFL
      RETURN
      END
      SUBROUTINE COPYER(M,A,B,NCOPY)
```

```
C
      DIMENSION FORMS(2),FORM(3)
      DIMENSION A(20),B(10),SP(2)
      DATA FORM/8H(       ,8H       ,8HX,1F0.0)/
      DATA SP/8H        ,8H        /
C
      CALL COPY8(A(1),SP(1))
      IF(NCOPY+1)1,2,3
1     NC=-1
      NCOPY=-NCOPY-1
      GO TO 5
2     NCOPY=81
3     NC=0
5     K=1
      N=1
      DO 60 I=1,80
      J=1
      CALL COMP(J,B(1),I,SP(1),1)
      IF(J)10,10,20
10    CALL COPY(1,A(N),K,B(1),I)
      IF(K.GE.M)GO TO 50
      K=K+1
      GO TO 60
20    IF(K-1)60,60,30
30    L=M-K+1
      IF(L)50,50,40
40    CALL COPY(L,A(N),K,SP(1),1)
50    K=1
      N=N+M/8
      IF(N-NCOPY)60,60,70
60    CONTINUE
70    IF(NC.GE.0)GO TO 110
      I=I-1
      WRITE(8,100)I
100   FORMAT(I8)
      READ(8,101)FORM(2)
101   FORMAT(A8)
      READ(7,FORM)A(NCOPY+1)
110   NCOPY=N-1
      RETURN
      END
      SUBROUTINE CHECKERR(I)
```

```
C       WHICH MAY HAVE CAUSED THE PROGRAM TO STOP.
C
        DIMENSION BUFFER(10)
C
        COMMON /SWITCH/ JSWTCH
C
        READ(7,5)BUFFER
5       FORMAT(10A8)
        WRITE(2,10)BUFFER
10      FORMAT(////25X,25H *****    E R R O R   *****  ////
       *10X,65H THE PROGRAM HAS STOPPED DUE VERY PROBABLY TO A WRONG ITEM
       *ON THE  /10X,17H FOLLOWING CARD    // 1X,10A8//
       *10X,60H CHECK THE PRESENCE OF ANY ALPHANUMERIC STRING OF CHARACTER
       *S  /10X,53H IT MAY BE A WRONG UNIT/STREAM/COMPONENT/COMMAND NAME
       *.10X,27H WHICH HAS CAUSED THE ERROR    // )
C
        IF(JSWTCH.EQ.0)GO TO 20
        WRITE(2,15)
15      FORMAT(//
       *10X,61H PLEASE CORRECT THE ERROR AND RE-INPUT THE WRONG CARD AND T
       *HE/10X,51H FOLLOWING ONES ONLY WITH THE APPROPRIATE COMMANDS.
       */10X,56H THE STATE OF YOUR STORED DATA IS GIVEN IN THE FOLLOWING /
       *10X,10H DATA ECHO  //
       *25X,22H *****  S T O P  *****  ////)
C
        INDEX=0
        NWFILE=4
        CALL FILER(NWFILE,INDEX)
        GO TO 30
C
20        WRITE(2,25)
25      FORMAT( //
       *10X,60H PLEASE RE-INPUT YOUR WHOLE SET OF DATA AFTER CORRECTING TH
       *E/10X,7H ERROR         ///
       *25X,22H *****  S T O P  *****  ////)
30      PAUSE ER
        END
        FINISH
```

Program PEETPACKSUBS

```
      PROGRAM(PEETPACK)
      INPUT 1 = CR0
      OUTPUT 2 = LP0
      OUTPUT 3 = LP1
      OUTPUT 4 = CP0
      TRACE 2,500
      COMPRESS INTEGER AND LOGICAL
      END
      MASTER PEETPACK
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),
     *NSLMAX,NL,NW
      COMMON /CCOMP/NC,NELIST(30,2),NLIST,NLOOPS,KONVRG,NOGO,NPS,KPS(10)
      COMMON /CMATRX/ KPM(30,8),SMATRX(400),NEMAT(30,2),EEN(500),
     1KSFLAG(30),NSLMX,NCOLPM,TOL,KSEM(30,3)
      COMMON /CNAME/ SNAME(30),ENAME(30),RNAME(30),VNAME(60),
     1MODULE(30),TITLE(10),NSMAX,NEMAX,NPAR,NVNAME,ITRACE,NPROP,NUSRPPT
     2,IRT,NW2
      COMMON /MINI/ IERR,E(30)
C
C
C     EXTERNAL CHECKERR
C
C     CALL FTRAP(CHECKERR)
C
C
C ***  PEETPACK  IS DESIGNED TO ACCEPT ONLY 30 UNITS AND 30 STREAMS.
C ***  TO GENERALISE ITS APPLICATION ONE OF THE TWO FOLLOWING
C ***  IMPROVEMENTS MAY BE BROUGHT TO IT :
C ***  1 -  CHANGE ALL DIMENSION AND COMMON STATEMENTS TO ACCEPT
C ***       MORE UNITS AND STREAMS BY INCREASING THEIR SIZES.
C ***  2 -  CHANGE THE WAY ALL DATA ARE TRANSFERRED THROUGH THE PACKAGE
C ***       BY THE COMMON STATEMENTS.  THIS CAN BE ACHIEVED EITHER
C ***       RESTRUCTURING THE ARRAYS SO THAT THEY SHARE COMMON
C ***       BOUNDARIES MOVEABLE AT WILL FROM WITHIN TO ACCOMODATE
C ***       A VARIABLE NUMBER OF STREAMS AND UNITS
C ***       OR BY INTRODUCING DYNAMIC DIMENSIONING THROUGHOUT THE
C ***       PACKAGE.
C ***  SUCH IMPROVEMENTS WERE NOT INITIATED BECAUSE OF LACK OF TIME.
C
C
```

```
      NW=2
      NW2=3
      NWFILE=4
      CALL DREAD(NR)
      IF(IERR.EQ.0)GO TO 10
      CALL MINIMAX
      IERR=0
      GO TO 20
10    CALL COMPUT(NW)
20    IOPTN=0
      CALL RESULT(IOPTN,NW)
      PAUSE FILEA
      CALL DPRINT(NWFILE)
      STOP
      END
C
      FUNCTION PUREVF(I,TRANSFER)
C
      DIMENSION TRANSFER(30)
C
      TIN=TRANSFER(2)
      HIN=TRANSFER(4)
      BP=BOILPT(I,TRANSFER(3))
      TRANSFER(2)=BP
      TRANSFER(5)=3.0
      CALL ENTHALPY(TRANSFER)
      HV=TRANSFER(4)
      TRANSFER(5)=2.0
      CALL ENTHALPY(TRANSFER)
      HL=TRANSFER(4)
      PUREVF=(HIN-HL)/(HV-HL)
      IF(PUREVF.LT.0.0)PUREVF=0.0
      IF(PUREVF.GT.1.0)PUREVF=1.0
      TRANSFER(5)=PUREVF
      TRANSFER(2)=TIN
      TRANSFER(4)=HIN
      RETURN
      END
      FINISH
```

Program EXECUTIVESUBS

```
      SEGMENTS(FXXX)
      INPUT 1 = CR0
      OUTPUT 2 = LP0
      OUTPUT 4 = CP0
      OUTPUT 6 = LP1
      COMPRESS INTEGER AND LOGICAL
      TRACE 2,500
      END
      SUBROUTINE DREAD (NR)
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
     *,NW
      COMMON /CCOMP/NC,NELIST(30,2),NLIST,NLOOPS,KONVRG,NOGO,NPS,KPS(10)
      COMMON /CMATRX/ KPM(30,8),SMATRX(400),NEMAT(30,2),EEN(500),
     1KSFLAG(30),NSLMX,NCOLPM,TOL,KSEM(30,3)
      COMMON /CNAME/ SNAME(30),ENAME(30),RNAME(30),VNAME(60),
     1MODULE(30),TITLE(10),NSMAX,NEMAX,NPAR,NVNAME,ITRACE,NPROP,NUSRPPT
     2,IRT,NW2
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS,NCTOT
     *,MAXNSL,NIL,NW1
      COMMON /PPTEXTRA/ PRTEXTRA(10,35),NPROPX
      COMMON /MINI/ IERR,E(30)
C
      NW1=NW
10    FORMAT(16I5)
20    FORMAT(10A8)
40    FORMAT(5E15.7)
      READ(NR,20)TITLE
      READ(NR,10)NEMAX,NSMAX,NOCOMP,NSLMAX,NPAR,NCOLPM,KONVRG,NPS,
     1ITRACE,NLOOPS,NLIST,NVNAME,NPROP,NOGO,IUNITS,NUSRPPT,NCTOT
     2,NPROPX,IRT,IERR
      MAXNSL=NSLMAX
      NSLMX=NSLMAX
      NL=NSLMAX-NOCOMP
      NIL=NL
      READ(NR,40)TOL
      READ(NR,20)(SNAME(I),I=1,NSMAX)
      READ(NR,20)(ENAME(I),I=1,NEMAX)
      IF(IRT.EQ.0)GO TO 12
      READ(NR,20)(RNAME(I),I=1,NEMAX)
      READ(NR,10)(MODULE(I),I=1,NEMAX)
```

```
12      READ(NR,20)(VNAME(I),I=1,NVNAME)
        IF(NOCOMP)18,18,13
13      DO 14 I=1,NOCOMP
        READ(NR,20)(PROPRT(I,J),J=1,2)
        READ(NR,40)(PROPRT(I,J),J=3,NPROP)
14      READ(NR,40)(PRTEXTRA(I,J),J=1,NPROPX)
        IF(NUSRPPT)18,18,16
16      DO 17 I=1,NCTOT
17      READ(NR,40)(UPPT(I,J),J=1,NUSRPPT)
18      READ(NR,10)((KPM(I,J),J=1,NCOLPM),I=1,NEMAX)
        READ(NR,10)((KSEM(I,J),J=1,3),I=1,NSMAX)
70      READ(NR,40)(SMATRX(I),I=1,NSMAX*NSLMAX)
        IF(NPAR.EQ.0)GO TO 80
        READ(NR,40)(EEN(I),I=1,NPAR)
        READ(NR,10)((NEMAT(I,J),J=1,2),I=1,NEMAX)
80      IF(NLIST)95,95,90
90      READ(NR,10)((NELIST(I,J),J=1,2),I=1,NLIST)
        READ(NR,10)(KSFLAG(I),I=1,NSMAX)
95      IF(NPS)100,100,96
96      READ(NR,10)(KPS(I),I=1,NPS)
100     IF(IERR.EQ.0)RETURN
        READ(NR,40)(E(I),I=1,NSMAX)
        RETURN
        END
        SUBROUTINE GETPARAM(MOD,IE,UPARAM)
C
        COMMON /CNAME/ SNAME(30),ENAME(30),RNAME(30),VNAME(60),
       1MODULE(30),TITLE(10),NSMAX,NEMAX,NPAR,NVNAME,ITRACE,NPROP,NUSRPPT
        COMMON /CMATRX/ KPM(30,8),SMATRX(400),NEMAT(30,2),EEN(500),
       1KSFLAG(30),NSLMX,NCOLPM,TOL,KSEM(30,3)
        COMMON /CPARAM/ M,N
        DIMENSION UPARAM(30)
C
        IF(MOD.GT.0)GO TO 70
        M=NEMAT(IE,1)
        N=NEMAT(IE,2)
        IF(N.EQ.0)RETURN
        DO 40 I=1,30
40      UPARAM(I)=0.0
        DO 50 I=1,N
50      UPARAM(I)=EEN(M+I-1)
```

```
      DO 90 I=1,N
90    FEN(M+I-1)=UPARAM(I)
      RETURN
      END
      SUBROUTINE EQCALL(NE)
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
      COMMON /CNAME/ SNAME(30),ENAME(30),RNAME(30),VNAME(60),
     1MODULE(30),TITLE(10),NSMAX,NEMAX,NPAR,NVNAME,ITRACE,NPROP,NUSRPPT
C
      I=MODULE(NE)
      GO TO (5,10,20,30,40,50,60,70,80,90,100,110,120,130,140,150,160,
     1170,180,190,200),I
5     CALL MASSMIX
      RETURN
10    CALL MIXER
      RETURN
20    CALL SPLITTER
      RETURN
30    CALL SPLITMIX
      RETURN
40    CALL FLASHER
      RETURN
50    CALL PURIFYER
      RETURN
60    CALL SETBP
      RETURN
70    CALL SETDP
      RETURN
80    CALL HEATER
      RETURN
90    CALL HEATEXCH
      RETURN
100   CALL PUMP
      RETURN
110   CALL COMPRESR
      RETURN
120   CALL VALVE
      RETURN
130   CALL DISCOL
      RETURN
```

```
140   CALL REACTOR
      RETURN
150   CALL UNIT1
      RETURN
160   CALL UNIT2
      RETURN
170   CALL UNIT3
      RETURN
180   CALL UNIT4
      RETURN
190   CALL UNIT5
      RETURN
200   CALL UNIT6
      RETURN
210   CALL UNIT7
      RETURN
      END
      SUBROUTINE TEST(NE,LIMIT)
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
      COMMON /CMATRX/ KPM(30,8),SMATRX(400),NEMAT(30,2),EEN(500),
     1KSFLAG(30),NSLMX,NCOLPM,TOL,KSEM(30,3)
C
      LIMIT=0
      DO 40 I=1,NOUT
      J=NIN+I+1
      J=-KPM(NE,J)
      J=(J-1)*NSLMAX
      DO 30 K=1,NSLMAX
      IF(ABS(STRMO(I,K))-0.00001)30,30,10
10    X=ABS((SMATRX(J+K)-STRMO(I,K))/STRMO(I,K))
      IF(X.LE.TOL)GO TO 30
20    LIMIT=1
      GO TO 50
30    CONTINUE
40    CONTINUE
50    RETURN
      END
      SUBROUTINE PUTSTM(NE)
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
```

```
C
      DO 20 I=1,NOUT
      J=NIN+I+1
      J=-KPM(NE,J)
      J=(J-1)*NSLMAX
      DO 10 L=1,NSLMAX
10    SMATRX(J+L)=STRMO(I,L)
20    CONTINUE
      N=NEMAT(NE,2)
      IF(N)40,40,25
25    M=NEMAT(NE,1)
      DO 30 I=1,N
30    EEN(M+I-1)=PARAM(I)
40    RETURN
      END
      SUBROUTINE PROMOT1(NS,IS,LOOP)
C
      COMMON /CONVRG/ CONVY(5,30),CONVX(5,15)
      COMMON /CMATRX/ KPM(30,8),SMATRX(400),NEMAT(30,2),EEN(500),
     1KSFLAG(30),NSLMAX,NCOLPM,TOL,KSEM(30,3)
C
      NS1=(NS-1)*NSLMAX
      IF(LOOP.GT.2)GO TO 20
      K=(LOOP-1)*NSLMAX
      DO 10 I=1,NSLMAX
      J=K+I
      CONVX(IS,I)=SMATRX(NS1+I)
10    CONVY(IS,J)=SMATRX(NS1+I)
      RETURN
20    DO 40 I=1,NSLMAX
      J=NSLMAX+I
      X=SMATRX(NS1+I)
      Y=X+CONVY(IS,I)-CONVX(IS,I)-CONVY(IS,J)
      IF(ABS(Y).LT.0.001)GO TO 30
      SMATRX(NS1+I)=(X*CONVY(IS,I)-CONVX(IS,I)*CONVY(IS,J))/Y
30    CONVX(IS,I)=X
      CONVY(IS,I)=CONVY(IS,J)
40    CONVY(IS,J)=SMATRX(NS1+I)
      RETURN
      END
      SUBROUTINE PROMOT2(NS,IS,LOOP)
```

```
C
      COMMON /CONVRG/ CONVST(5,30)
      COMMON /CMATRX/ KPM(30,8),SMATRX(400),NEMAT(30,2),EEN(500),
     1KSFLAG(30),NSLMAX,NCOLPM,TOL,KSEM(30,3)
C
      NS1=(NS-1)*NSLMAX
      J=MOD(LOOP,3)
      IF(J)30,30,10
10    J=(J-1)*NSLMAX
      DO 20 I=1,NSLMAX
20    CONVST(IS,J+I)=SMATRX(NS1+I)
      RETURN
30    DO 50 I=1,NSLMAX
      X=2.*CONVST(IS,NSLMAX+I)-SMATRX(NS1+I)-CONVST(IS,I)
      IF(ABS(X)-0.0001)50,50,40
40    X=(SMATRX(NS1+I)-CONVST(IS,NSLMAX+I))/X
      SMATRX(NS1+I)=SMATRX(NS1+I)*(1.+X)-CONVST(IS,NSLMAX+I)*X
50    CONTINUE
      RETURN
      END
      SUBROUTINE RESULT(IOPTN,NW)
C
      COMMON /CMATRX/ KPM(30,8),SMATRX(400),NEMAT(30,2),EEN(500),
     1KSFLAG(30),NSLMAX,NCOLPM,TOL,KSEM(30,3)
      COMMON /CNAME/ SNAME(30),ENAME(30),RNAME(30),VNAME(60),
     1MODULE(30),TITLE(10),NSMAX,NEMAX,NPAR,NVNAME,ITRACE,NPROP,NUSRPPT
C
      IF(IOPTN)2,2,4
2     WRITE(NW,100)
      GO TO 6
4     WRITE(NW,105)IOPTN
6     N=NSMAX/3
      M=NSMAX-N*3
      IF(N)40,40,10
10    K=-2
      DO 30 I=1,N
      K=K+3
      K1=K+2
      WRITE(NW,110)(SNAME(J),J=K,K1)
      J1=-1
      DO 20 J=1,NSLMAX
```

```
30     CONTINUE
40     IF(M)90,90,50
50     N=N*3+1
       WRITE(NW,110)(SNAME(J),J=N,NSMAX )
       J1=-1
       DO 60 J=1,NSLMAX
       J1=J1+2
60     WRITE(NW,120)VNAME(J1),VNAME(J1+1),(SMATRX((L-1)*NSLMAX+J),
      ,L=N,NSMAX)
70     IF(IOPTN)72,72,74
72     WRITE(NW,130)
       GO TO 76
74     WRITE(NW,135)
76     DO 80 I=1,NEMAX
       M=NEMAT(I,1)
       N=NEMAT(I,2)+M-1
       IF(N.GT.0)GO TO 78
       WRITE(NW,150)ENAME(I)
       GO TO 80
78     WRITE(NW,140)ENAME(I),(EEN(J),J=M,N)
80     CONTINUE
90     RETURN
100    FORMAT(1H1//10X,20H FINAL STREAM MATRIX/11X,19(1H*)/)
105    FORMAT(1H1//10X,24H STREAM MATRIX FOR LOOP ,I2/11X,23(1H*)//)
110    FORMAT(//30X,3(4X,A8,3X)/)
120    FORMAT(10X,2A8,4X,3F15.5)
130    FORMAT(1H1//10X,33H FINAL EQUIPMENT PARAMETER MATRIX/
      111X,32(1H*)//8X,10H UNIT NAME,10X,11H PARAMETERS  /)
135    FORMAT(1H1//10X,17H EQUIPMENT MATRIX/11X,16(1H*)//8X,10H UNIT NAME
      1,10X,11H PARAMETERS  /)
140    FORMAT(/10X,A8,2X,4F15.5,6(/20X,4F15.5))
150    FORMAT(/10X,A8,6X,18H HAS NO PARAMETERS )
       END
       SUBROUTINE DPRINT(NWFILE)
C
       COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
       COMMON /CMATRX/ KPM(30,8),SMATRX(400),NEMAT(30,2),EEN(500),
      1KSFLAG(30),NSLMX,NCOLPM,TOL,KSEM(30,3)
       COMMON /CNAME/ SNAME(30),ENAME(30),RNAME(30),VNAME(60),
      1MODULE(30),TITLE(10),NSMAX,NEMAX,NPAR,NVNAME,ITRACE,NPROP,NUSRPPT
      2,IRT
```

```
      COMMON /PROPS/ PROPRT(10,25),USERPPT(10,10),NOCOMP,IUNITS,NCTOT
      COMMON /CCOMP/NC,NELIST(30,2),NLIST,NLOOPS,KONVRG,NOGO,NPS,KPS(10)
      COMMON /PPTEXTRA/ PRTEXTRA(10,35),NPROPX
      COMMON /MINI/ IERR,E(30)
C
      WRITE(NWFILE,100)TITLE
      WRITE(NWFILE,110)NEMAX,NSMAX,NOCOMP,NSLMAX,NPAR,NCOLPM,KONVRG,
     1NPS,ITRACE,NLOOPS,NLIST,NVNAME,NPROP,NOGO,IUNITS,NUSRPPT,NCTOT
     2,NPROPX,IRT,IERR
      WRITE(NWFILE,130)TOL
      WRITE(NWFILE,100)(SNAME(I),I=1,NSMAX)
      WRITE(NWFILE,100)(ENAME(I),I=1,NEMAX)
      IF(IRT.EQ.0)GO TO 5
      WRITE(NWFILE,100)(RNAME(I),I=1,NEMAX)
      WRITE(NWFILE,110)(MODULE(I),I=1,NEMAX)
5     WRITE(NWFILE,100)(VNAME(I),I=1,NVNAME)
      IF(NOCOMP)20,20,10
10    DO 12 I=1,NOCOMP
      WRITE(NWFILE,100)(PROPRT(I,J), J=1,2)
      WRITE(NWFILE,130)(PROPRT(I,J),J=3,NPROP)
12    WRITE(NWFILE,130)(PRTEXTRA(I,J),J=1,NPROPX)
      IF(NUSRPPT)20,20,14
14    DO 15 I=1,NCTOT
15    WRITE(NWFILE,130)(USERPPT(I,J),J=1,NUSRPPT)
20    WRITE(NWFILE,110)((KPM(I,J),J=1,NCOLPM),I=1,NEMAX)
      WRITE(NWFILE,110)((KSEM(I,J),J=1,3),I=1,NSMAX)
      WRITE(NWFILE,130)(SMATRX(I),I=1,NSLMAX*NSMAX)
      IF(NPAR.EQ.0)GO TO 25
      WRITE(NWFILE,130)(EEN(I),I=1,NPAR)
      WRITE(NWFILE,110)((NEMAT(I,J),J=1,2),I=1,NEMAX)
25    IF(NLIST)40,40,30
30    WRITE(NWFILE,110)((NELIST(I,J),J=1,2),I=1,NLIST)
      WRITE(NWFILE,110)(KSFLAG(I),I=1,NSMAX)
40    IF(NPS)60,60,50
50    WRITE(NWFILE,110)(KPS(I),I=1,NPS)
60    IF(IERR.EQ.0)RETURN
      WRITE(NWFILE,130)(E(I),I=1,NSMAX)
      RETURN
100   FORMAT(10A8)
110   FORMAT(16I5)
130   FORMAT(5E15.7)
```

```
C
      DIMENSION TRANSFER(30),SNS(12)
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
      COMMON /CMATRX/ KPM(30,8),SMATRX(400),NEMAT(30,2),EEN(500),
     1KSFLAG(30),NSLMX,NCOLPM,TOL,KSEM(30,3)
      COMMON /CNAME/ SNAME(30),ENAME(30),RNAME(30),VNAME(60),
     1MODULE(30),TITLE(10),NSMAX,NEMAX,NPAR,NVNAME,ITRACE ,NPROP,NUSRPPT
C
      NIN=0
      NOUT=0
      DO 5 I=1,6
      STRMI(I,1)=0.
5     STRMI(I,4)=0.
      DO 50 I=2,NCOLPM
      L=KPM(NE,I)
      IF(L)30,60,10
10    CALL COPY8(SNS(I-1),SNAME(L))
      L=(L-1)*NSLMAX
      NIN=NIN+1
      DO 20 J=1,NSLMAX
20    STRMI(NIN,J)=SMATRX(L+J)
      GO TO 50
30    L=-L
      CALL COPY8(SNS(I-1),SNAME(L))
      L=(L-1)*NSLMAX
      NOUT=NOUT+1
      DO 40 J=1,NSLMAX
40    STRMO(NOUT,J)=SMATRX(L+J)
50    CONTINUE
60    M=NEMAT(NE,1)
      N=NEMAT(NE,2)
65    DO 66 I=1,30
66    PARAM(I)=0.0
      IF(N.LE.0)GO TO 80
68    DO 70 I=1,N
70    PARAM(I)=EEN(M+I-1)
80    DO 140 I=1,NIN
      IF(ABS(STRMI(I,4)).LT.0.01.AND.NSLMAX.GT.5)GO TO 90
      GO TO 110
90    DO 100 J=1,NSLMAX
100   TRANSFER(J)=STRMI(I,J)
```

```
      TRANSFER(5)=-1.
      CALL ENTHALPY(TRANSFER)
      STRMI(I,4)=TRANSFER(4)
      STRMI(I,5)=TRANSFER(5)
110   IF(STRMI(I,1))120,120,140
120   WRITE(NW,130)SNS(I),ENAME(NE)
130   FORMAT(/10X,26H **WARNING** INPUT STREAM ,A8,9H TO UNIT ,2X,A8/
     122X,14H BEARS NO FLOW/)
140   CONTINUE
      IF(ITRACE.NE.0.AND.MODE.EQ.0)GO TO 155
145   WRITE(NW,150)RNAME(NE),ENAME(NE)
150   FORMAT(//10X,9H ROUTINE ,A8,24H CALLED TO SIMULATE UNIT ,2X,A8/)
155   IF(ITRACE)240,160,240
160   NLM=NIN/3
      NM=NIN-NLM*3
      WRITE(NW,165)
165   FORMAT(10X,17H INPUT STREAMS : )
      IF(NLM)220,220,170
170   I1=-2
      DO 210 I=1,NLM
      I1=I1+3
      I2=I1+2
      WRITE(NW,180)(SNS(K),K=I1,I2)
180   FORMAT(30X,3(7X,A8,1X))
      J1=-1
      DO 190 J=1,NSLMAX
      J1=J1+2
190   WRITE(NW,200)VNAME(J1),VNAME(J1+1),(STRMI(L,J),L=I1,I2)
200   FORMAT(10X,2A8,4X,3F15.5)
210   CONTINUE
220   IF(NM)240,240,225
225   NLM=NLM*3+1
      WRITE(NW,180)(SNS(K),K=NLM,NIN)
      J1=-1
      DO 230 I=1,NSLMAX
      J1=J1+2
230   WRITE(NW,200)VNAME(J1),VNAME(J1+1),(STRMI(L,I),L=NLM,NIN)
240   CALL EQCALL(NE)
      IF(MODE)245,245,250
245   IF(ITRACE)400,250,400
250   WRITE(NW,260)
```

```
      NM=NOUT-NLM*3
      IF(NLM)310,310,280
280   I1=NIN-2
      DO 300 I=1,NLM
      I1=I1+3
      I2=I1+2
      WRITE(NW,180)(SNS(K),K=I1,I2)
      I3=I1-NIN
      I4=I2-NIN
      J1=-1
      DO 290 J=1,NSLMAX
      J1=J1+2
290   WRITE(NW,200)VNAME(J1),VNAME(J1+1),(STRMO(L,J),L=I3,I4)
300   CONTINUE
310   IF(NM)400,400,315
315   NLM=NLM*3+1
      I1=NLM+NIN
      I2=NOUT+NIN
      WRITE(NW,180)(SNS(K),K=I1,I2)
      J1=-1
      DO 320 I=1,NSLMAX
      J1=J1+2
320   WRITE(NW,200)VNAME(J1),VNAME(J1+1),(STRMO(L,I),L=NLM,NOUT)
400   RETURN
      END
      SUBROUTINE COMPUT(NW)
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
      COMMON /CCOMP/NC,NELIST(30,2),NLIST,NLOOPS,KONVRG,NOGO,NPS,KPS(10)
      COMMON /CMATRX/ KPM(30,8),SMATRX(400),NEMAT(30,2),EEN(500),
     1KSFLAG(30),NSLMX,NCOLPM,TOL,KSEM(30,3)
      COMMON /CNAME/ SNAME(30),ENAME(30),RNAME(30),VNAME(60),
     1MODULE(30),TITLE(10),NSMAX,NEMAX,NPAR,NVNAME,ITRACE,NPROP,NUSRPPT
     2,IRT,NW2
C
      WRITE(NW,300)TITLE
      IERR=0
      MODE=0
      LOOPN=1
      LOOP=1
      NELOOP=0
```

```
      KTEST=0
5     NC=0
10    NC=NC+1
      IF(NC-NLIST)20,20,200
20    NE=NELIST(NC,1)
      IF(MODE)25,25,30
25    IF(NELIST(NC,2)-2)30,40,40
30    CALL LOADST(NE,NW)
      CALL PUTSTM(NE)
      GO TO 10
40    NELOOP=NELOOP+1
      IF(NELOOP.GT.1.OR.LOOP.GT.1)GO TO 48
      IF(KONVRG.LT.1)GO TO 48
      IS=0
      DO 45 I=2,NCOLPM
      IF(KPM(NE,I).LE.0)GO TO 45
42    NS=KPM(NE,I)
      IF(KSFLAG(NS).NE.2)GO TO 45
      IS=IS+1
      GO TO (43,44),KONVRG
43    CALL PROMOT1(NS,IS,1)
      GO TO 45
44    CALL PROMOT2(NS,IS,1)
45    CONTINUE
48    CALL LOADST(NE,NW)
      CALL TEST(NE,LTEST)
      IF(LTEST)60,60,50
50    KTEST=1
60    CALL PUTSTM(NE)
      IF(NELIST(NC,2)-2)10,10,70
70    IF(KTEST)160,160,80
80    LOOP=LOOP+1
      IF(ITRACE)85,85,81
81    LP1=LOOP-1
      IF(MOD(LP1,ITRACE))82,82,85
82    CALL RESULT(LOOPN,NW)
85    IF(LOOP-NLOOPS)90,90,110
90    NC=NC-NELOOP
      IF(KONVRG)105,105,95
95     NC=NC+1
      NE=NELIST(NC,1)
```

```
      IF(KPM(NE,I))104,104,96
96    NS= KPM(NE,I)
      IF(KSFLAG(NS)-2)103,97,103
97    IS=IS+1
      GO TO (100,101),KONVRG
100   CALL PROMOT1(NS,IS,LOOP)
      GO TO 103
101   CALL PROMOT2(NS,IS,LOOP)
103   CONTINUE
104   NC=NC-1
105   NELOOP=0
      KTEST=0
      GO TO 10
110   IF(NOGO)120,120,140
120   WRITE(NW,130)LOOPN,LOOP
130   FORMAT(////25X,6H LOOP ,I4,23H FAILED TO CONVERGE IN ,I4,6H LOOPS/
     110X,60H LAST RESULTS TAKEN AS FINAL VALUES AND SIMULATION CONTINUE
     2D//)
      LOOP=1
      NELOOP=0
      KTEST=0
      LOOPN=LOOPN+1
      GO TO 10
140   WRITE(NW,150)LOOPN,LOOP
150   FORMAT(///25X,6H LOOP ,I4,23H FAILED TO CONVERGE IN ,I4,6H LOOPS//
     125X,22H SIMULATION ABANDONNED//)
      LOOPN=0
      JERR=1
      GO TO 185
160   IF(ITRACE)190,170,170
170   WRITE(NW,180)LOOPN,LOOP
180   FORMAT(///25X,6H LOOP ,I2,14H CONVERGED IN ,I2,6H LOOPS /)
185   CALL RESULT(LOOPN,NW)
      IF(JERR.EQ.1)PAUSE
190   LOOP=1
      LOOPN=LOOPN+1
      NELOOP=0
      GO TO 10
200   IF(MODE.GT.0)GO TO 220
      WRITE(NW,240)
      MODE =1
```

```
      GO TO 5
220   WRITE(NW,210)
210   FORMAT(///25X,23H SIMULATION SUCCESSFUL //25X,15H RESULTS STORED/)
240   FORMAT(1H1//30X,'FINAL RUN THROUGH THE PLANT'//)
230   RETURN
300   FORMAT(1H1//20X,'PRINT OUT FOR THE SIMULATION PROBLEM ENTITLE'
     */20X,45('*')//5X,10A8/5X,80('*')//)
      END
      SUBROUTINE MINIMAX
      COMMON /CMATRX/KPM(30,8),SMATRX(400),NEMAT(30,2),EEN(500),
     1KSFLAG(30),NSLMX,NCOLPM,TOL,KSEM(30,3)
      COMMON /CNAME/SNAME(30),FNAME(30),RNAME(30),VNAME(60),
     1MODULE(30),TITLE(10),NSMAX,NEMAX,NPAR,NVNAME,ITRACE,NPROP,NUSRPPT
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
      COMMON /MINI/ IERR,E(30)
      DIMENSION A(30,30),B(30,30),C(30,30),D(30,60),Z(30)
C
      DIMENSION KEFLAG(30)
      KTEST=0
      DO 1 I=1,30
      DO 1 J=1,30
      A(I,J)=0.0
      B(I,J)=0.0
      C(I,J)=0.0
      D(I,J)=0.0
1     D(I,J+30)=0.0
C
C *** SET ALL FLOW METER ERRORS
C
      EM=E(1)
      DO 5 I=2,NSMAX
      IF(EM.LT.E(I))EM=E(I)
5     CONTINUE
      EM=EM*2.0
      DO 6 I=1,NSMAX
      IF(E(I).EQ.0.0)E(I)=EM
6     CONTINUE
      GO TO 500
7     CONTINUE
C
C *** CREATE MATRIX A, THE INCIDENCE MATRIX
```

```
      NEMAX1=NEMAX
      NE=0
C
      DO 40 I=1,NEMAX
      NE=NE+1
      IF(MODULE(I).EQ.10)GO TO 25
      IF(MODULE(I).EQ.14)GO TO 30
9     DO 20 J=2,NCOLPM
      K=KPM(I,J)
      IF(K)10,40,15
10    K=-K
      A(NE,K)=-1.0
      GO TO 20
15    A(NE,K)=1.0
20    CONTINUE
      GO TO 40
C *** DECOUPLING OF THE TWO INPUTS TO THE HEAT EXCHANGER
25    K=KPM(I,2)
      A(NE,K)=1.0
      K=-KPM(I,4)
      A(NE,K)=-1.0
      NE=NE+1
      K=KPM(I,3)
      A(NE,K)=1.0
      K=-KPM(I,5)
      A(NE,K)=-1.0
      GO TO 40
30    IF(KPM(I,4).LE.0)GO TO 9
C *** DECOUPLING OF THE COLUMN MODEL AND THE REBOILER MODEL
      K=KPM(I,2)
      A(NE,K)=1.0
      K=KPM(I,3)
      A(NE,K)=1.0
      K=-KPM(I,5)
      A(NE,K)=-1.0
      K=-KPM(I,6)
      A(NE,K)=-1.0
      NE=NE+1
      K=KPM(I,4)
      A(NE,K)=1.0
      K=-KPM(I,7)
```

```
      A(NE,K)=-1.0
40    CONTINUE
C
      NEMAX=NE
C
      DO 2010 I=1,NEMAX
2010  WRITE(2,2020)(A(I,J),J=1,NSMAX)
2020  FORMAT(10X,'A',20F4.1)
C
C *** CREATE MATRIX C, THE ERROR VARIANCE MATRIX
C
      K=-NSLMAX
      DO 60 I=1,NSMAX
50    K=K+NSLMAX
      C(I,I)=0.5*(SMATRX(K+1)*E(I))**2
60    CONTINUE
      WRITE(2,2030)(C(I,I),I=1,NSMAX)
2030  FORMAT(10X,'C',10F10.5)
C
C *** MULTIPLY MATRIX A TRANSPOSE BY MATRIX C AND SET THE RESULTS IN MATRIX B
C *** B=C*AT
C
      DO 70 I=1,NSMAX
      DO 70 J=1,NSMAX
70    B(I,J)=C(I,I)*A(J,I)
      DO 2040 I=1,NSMAX
2040  WRITE(2,2050)(B(I,J),J=1,NSMAX)
2050  FORMAT(10X,'B',10F10.5)
C
C *** MULTIPLY MATRIX B BY MATRIX A AND SET THE REULTS IN MATRIX D
C *** D=A*(C*AT)
C
      DO 80 I=1,NEMAX
      DO 80 J=1,NEMAX
      DO 80 K=1,NSMAX
80    D(I,J)=D(I,J)+A(I,K)*B(K,J)
      DO 2060 I=1,NEMAX
2060  WRITE(2,2070)(D(I,J),J=1,NEMAX)
2070  FORMAT(10X,'D',10F10.5)
C
C *** INVERT MATRIX D
```

```
      N2=NEMAX+NEMAX
      DO 130 I=1,NEMAX
130   D(I,I+NEMAX)=1.0
      DO 225 I=1,NEMAX
140   DIV=D(I,I)
      IF(DIV.EQ.0.0)GO TO 180
      DO 150 J=1,N2
150   D(I,J)=D(I,J)/DIV
      DO 170 K=1,NEMAX
      IF(K.EQ.I)GO TO 170
      FAD=D(K,I)
      DO 160 J=1,N2
160   D(K,J)=D(K,J)-FAD*D(I,J)
170   CONTINUE
      GO TO 225
180   I2=I+1
      DO 190 J=I2,NEMAX
      IF(A(J,I).NE.0.0)GO TO 210
190   CONTINUE
      WRITE(2,200)
200   FORMAT(/10X,'ERROR IN FORMAT 200 OF LAGRANGE'//)
      STOP
210   DO 220 K=I,N2
      XX=A(I,K)
      A(I,K)=A(J,K)
220   A(J,K)=XX
      GO TO 140
225   CONTINUE
      DO 230 I=1,NEMAX
      DO 230 J=1,NEMAX
      D(I,J)=D(I,J+NEMAX)
230   D(I,J+NEMAX)=0.0
C
C *** MULTIPLY MATRIX D BY MATRIX B AND SET THE RESULTS IN MATRIX D
C *** D=C*AT*(A*C*AT)**-1
C
      DO 240 I=1,NSMAX
      DO 240 J=1,NEMAX
240   C(I,J)=0.0
      DO 250 I=1,NSMAX
      DO 250 J=1,NEMAX
```

```
      DO 250 K=1,NEMAX
250   C(I,J)=C(I,J)+B(I,K)*D(K,J)
      DO 260 I=1,NSMAX
      DO 260 J=1,NEMAX
260   D(I,J)=C(I,J)
C
C *** MULTIPLY MATRIX A BY MATRIX D AND SET THE RESULTS IN MATRIX D
C *** D=(C*AT*(A*C*AT)**-1)*A
C
      DO 280 I=1,NSMAX
      DO 270 J=1,NEMAX
      Z(J)=D(I,J)
270   D(I,J)=0.0
      DO 280 J=1,NSMAX
      DO 280 K=1,NEMAX
280   D(I,J)=D(I,J)+Z(K)*A(K,J)
C
C *** OBTAIN THE NEW CORRECTIONS TO THE FLOW ESTIMATES
C *** X=X+C*AT*(A*C*AT)**-1*A*X
C
      IF(TOL.GT.0.0001)TOL=0.0001
      JJ=0
290   DO 295 I=1,NSMAX
295   Z(I)=0.0
      JJ=JJ+1
      L=-NSLMAX
      DO 310 I=1,NSMAX
      K=-NSLMAX
      DO 300 J=1,NSMAX
      K=K+NSLMAX
300   Z(I)=Z(I)+D(I,J)*SMATRX(K+1)
      L=L+NSLMAX
      XX=SMATRX(L+1)-Z(I)
      IF(ABS((XX-SMATRX(L+1))/XX).GT.TOL)KTEST=1
      SMATRX(L+1)=XX
310   CONTINUE
      IF(KTEST.EQ.0)GO TO 400
      KTEST=0
      WRITE(2,2200)JJ
2200  FORMAT(10X,'JJ,ITERATION',I3)
      GO TO 290
```

```
400   NEMAX=NEMAX1
C
      IF(IERR.EQ.2)RETURN
      IF(NSLMAX.LE.5)RETURN
      K=-NSLMAX
      DO 430 I=1,NSMAX
      K=K+NSLMAX
      DO 410 J=1,NSLMAX
410   Z(J)=SMATRX(K+J)
      CALL TOMOLE(Z)
      DO 420 J=1,NSLMAX
420   SMATRX(K+J)=Z(J)
430   CONTINUE
      RETURN
C
C *** CHECK THAT ALL STREAM FLOW RATES ARE KNOWN AND CALCULATE UNKNOWN ONES
C
500   KFLAG=0
      DO 665 IE=1,NEMAX
      IF(KEFLAG(IE).GT.0)GO TO 665
      WRITE(2,1000)ENAME(IE)
1000  FORMAT(10X,A8,' NOT YET CALCULATED')
      KFLAG=1
C *** CHECK ALL OUTPUTS
      DO 520 I=2,NCOLPM
      IF(KPM(IE,I))510,530,520
510   J=-KPM(IE,I)
      J=(J-1)*NSLMAX+1
      IF(SMATRX(J).EQ.0.0)GO TO 540
520   CONTINUE
530   KEFLAG(IE)=1
      GO TO 665
C *** CHECK ALL INPUTS
540   DO 550 I=2,NCOLPM
      IF(KPM(IE,I).LE.0)GO TO 560
      J=KPM(IE,I)
      J=(J-1)*NSLMAX+1
      IF(SMATRX(J).EQ.0.0)GO TO 665
550   CONTINUE
C
560   NIN=0
```

```
      NOUT=0
      DO 564 I=2,NCOLPM
      IF(KPM(IE,I))563,565,561
561   J=KPM(IE,I)
      J=(J-1)*NSLMAX
      NIN=NIN+1
      DO 562 K=1,NSLMAX
562   STRMI(NIN,K)=SMATRX(J+K)
      WRITE(2,3000)(STRMI(NIN,K),K=1,NSLMAX)
3000  FORMAT(10X,'STRMI',10F10.4)
      GO TO 564
563   NOUT=NOUT+1
564   CONTINUE
565   M=NEMAT(IE,1)
      N=NEMAT(IE,2)
      DO 566 I=1,30
566   PARAM(I)=0.0
      IF(N.LE.0)GO TO 568
      DO 567 I=1,N
567   PARAM(I)=EEN(M+I-1)
      WRITE(2,3010)(PARAM(I),I=1,N)
3010  FORMAT(10X,'PARAM',8F12.4)
568   IF(IERR.EQ.1)GO TO 600
      IF(NSLMAX.LE.5)GO TO 605
      WRITE(2,1020)
1020  FORMAT(10X,'CALL TO TOMOLE')
      DO 590 I=1,NIN
      DO 570 J=1,NSLMAX
570   Z(J)=STRMI(I,J)
      CALL TOMOLE(Z)
      DO 580 J=1,NSLMAX
580   STRMI(I,J)=Z(J)
590   CONTINUE
600   IF(NSLMAX.LE.5)GO TO 605
      DO 602 I=1,NIN
      IF(STRMI(I,4).NE.0.0)GO TO 602
      DO 601 J=1,NSLMAX
601   Z(J)=STRMI(I,J)
      Z(5)=-1.0
      CALL ENTHALPY(Z)
      STRMI(I,4)=Z(4)
```

```
605   CALL EQCALL(IE)
      WRITE(2,1030)RNAME(IE)
1030  FORMAT(10X,A8,' MODEL CALLED TO CALCULATE THE UNIT')
      KEFLAG(IE)=1
      DO 660 I=1,NOUT
      J=NIN+I+1
      J=-KPM(IE,J)
      J=(J-1)*NSLMAX
      IF(SMATRX(J+1).NE.0.0)GO TO 660
      DO 610 L=1,NSLMAX
610   Z(L)=STRMO(I,L)
      IF(IERR.EQ.1)GO TO 640
      IF(NSLMAX.LE.5)GO TO 640
      WRITE(2,1035)
1035  FORMAT(10X,'CALL TO TOMASS')
      CALL TOMASS(Z)
640   DO 650 L=1,NSLMAX
650   SMATRX(J+L)=Z(L)
660   CONTINUE
665   CONTINUE
      IF(KFLAG.GT.0)GO TO 500
C
      IF(IERR.EQ.2)GO TO 7
      IF(NSLMAX.LE.5)GO TO 7
      WRITE(2,1040)
1040  FORMAT(10X,'CALL TO TOMASS BEFORE LEAVING THE INITIATOR')
      K=-NSLMAX
      DO 690 I=1,NSMAX
      K=K+NSLMAX
      DO 670 J=1,NSLMAX
670   Z(J)=SMATRX(K+J)
      CALL TOMASS(Z)
      DO 680 J=1,NSLMAX
680   SMATRX(K+J)=Z(J)
690   CONTINUE
      GO TO 7
      END
      FINISH
```

Program THERMOSUBS

```
      LIST(LP)
      SEGMENTS(FXX4)
      INPUT 1 = CRJ
      OUTPUT 2 = LP0
      COMPRESS INTEGER AND LOGICAL
      TRACE 2,500
      END
      SUBROUTINE RKJ/TA(X,T)
C
      DIMENSION X(10),B(10)
      COMMON /PROPS/ PROPRT(10,25),USERPPT(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL
      COMMON /RK/BIGA(10),BIGB(10),AABAR,BBBAR,ABAR
C
      IF(IUNITS)1,1,2
1     TADD=460.
      R=10.7335
      GO TO 3
2     R=0.084778
      TADD=273.18
3     T=T+TADD
      DO 10 I=1,NOCOMP
      IF(PROPRT(I,12).GT.0.0)GO TO 6
      PROPRT(I,12)=.4278
      PROPRT(I,13)=.0867
6     B(I)=PROPRT(I,13)*PROPRT(I,6)/PROPRT(I,5)
      T1R=PROPRT(I,6)/T
      BIGA(I)=PROPRT(I,12)*T1R**2.5/PROPRT(I,5)
      BIGA(I)=SQRT(BIGA(I))
10    BIGB(I)=B(I)/T
      GO TO 35
      P=1./3.
      AII=0.
      AIJ=0.
      DO 30 I=1,NOCOMP
      DO 30 J=1,NOCOMP
      IF(I-J)20,30,20
20    TCIJ=SQRT(PROPRT(I,6)*PROPRT(J,6))
      VCIJ=((PROPRT(I,7)**P+PROPRT(J,7)**P)/2.)**3
      ZCIJ=.291-0.04*(PROPRT(I,9)+PROPRT(J,9))
      PCIJ=ZCIJ*R*TCIJ/VCIJ
      AIJ=AIJ+X(I)*X(J)*(PROPRT(I,12)+PROPRT(J,12))*R*R*TCIJ**2.5
```

```
35    CONTINUE
      BBBAR=0.
      BBAR=0.
      AABAR=0.
      ABAR=0.
      DO 40 I=1,NOCOMP
      AABAR=AABAR+BIGA(I)*X(I)
      BBBAR=BBBAR+BIGB(I)*X(I)
40    BBAR=BBAR+B(I)*X(I)
      GO TO 60
      DO 50 I=1,NOCOMP
50    AII=AII+PROPRT(I,12)*R*R*PROPRT(I,6)**2.5/PROPRT(I,5)*X(I)*X(I)
      ABAR=AIJ+AII
60    CONTINUE
      T=T-TADD
      RETURN
      END
      FUNCTION VAPRES(I,T)
C
      COMMON /PROPS/ PROPRT(10,25),U(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL
C
      IF(IUNITS.GT.0)GO TO 10
      T1=(T+460.)/1.8
      GO TO 20
10    T1=T+273.15
20    VAPRES=PROPRT(I,21)-(0.2185*PROPRT(I,20)/T1)
      VAPRES=EXP10(VAPRES)
      IF(IUNITS.GT.0)GO TO 30
      VAPRES=VAPRES/51.715
      RETURN
30    VAPRES=VAPRES/750.
      RETURN
      END
      FUNCTION BOILPT(I,P)
C
      COMMON /PROPS/ PROPRT(10,25),USERPPT(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX
C
      IF(IUNITS)10,10,20
```

```
10    PRES=P*51.715
      GO TO 30
20    PRES=P*750.
30    BOILPT=.2185*PROPRT(I,20)/(PROPRT(I,21)-ALOG10(PRES))
      IF(IUNITS)40,40,50
40    BOILPT=BOILPT*1.8-460.
      RETURN
50    BOILPT=BOILPT-273.15
      RETURN
      END
      SUBROUTINE HDATA(ISTATE,TR,PR,ZC,HOH,NW)
C
C     ISTATE = 0  :  SUBCOOLED LIQUID REGION
C     ISTATE = 1  :  SUPERHEATED VAPOUR REGION
C     ISTATE = 2  :  SATURATED LIQUID LINE
C     ISTATE = 3  :  SATURATED VAPOUR LINE
C
      PI=3.14159
      KALL=0
      IZC=IFIX((ZC-.2079)*100.)/2
5     IF(IZC)10,10,20
10    IZC=1
      GO TO 40
20    IF(IZC-4)40,40,30
30    IZC=4
40    IST1=ISTATE+1
      GO TO (300,50,800,500),IST1
C
C
C     SUPERHEATED VAPOUR PHASE
C
50    GO TO (180,170,130,100),IZC
C
100   Z=.20
      IF(TR.LE.0.75)GO TO  600
      IF(TR-8.)110,120,120
110   C1=166.*EXP(-5.16*TR)+.017
      C2=.62*EXP(-18.4*(TR-1.))+.05
      C3=EXP(38.8/TR-34.2)
      T1=TR-.75
      C4=.989*T1**1.63*EXP(-.3175*T1)
      C5=.215*EXP(-.1045*TR**4.735)
```

```
      X0=1.15-.314*(TR-8.)*(TR-8.)*(TR-8.)
      XM=(1.-.0001499*TR**9.1)*EXP(-1.297*TR))
      XM=XM/(.1879+1.0826*(TR-.65)**1.1726)
      GO TO 400
120   HOHTC=TR*(.053+TR*(-.00341+TR* .0000634 +.000508*PR))
     1+PR*(.13+PR*(-.00176+.0000258*PR+.000128*TR)-.02725*TR)-.2078
      IF(TR.GT.25.0)WRITE(NW,1110)
      GO TO 410
C
130   Z=.27
      T1=TR-1.
      C1=40.*EXP(-5.7*TR)/(TR-.81)+.01
      C2=.35*EXP(-26.2*T1)/TR+.31
      C3=17.25*EXP(-16.7*T1)+2.75
      T1=T1-1.
      C4=.6564+T1*(.444+T1*(-.0215+T1*(.061-.0404*T1)))
      IF(TR-2.25)140,140,150
140   C5=T1*(-.0697+T1*(.0734-.05333*T1*T1*T1))+.0125
      GO TO 160
150   C5=0.
160   T1=TR-.8
      C6=.00301*T1*EXP(-.87*T1*T1)
      T1=TR-4.
      X0=8.5+10.5*T1*T1
      XM=1./(.1052+1.2044*(TR-.4429)**2.135)
      IF(TR.GT.4.0)WRITE(NW,1110)
      GO TO 400
C
170   Z=.25
      IF(TR.LT.0.6)GO TO 700
      T1=1.-TR
      C1=.87*EXP(9.845*T1)+.023955*EXP(1.2405*TR)
      C2=.47148*EXP(39.35*T1)+.142
      C3=3918.*EXP(122.*T1)+2.
      C4=TR*(1.1938-.20726*TR)-.90649
      C5=3.*EXP(-2.52*TR)
      T1=TR-.805
      C6=.004277*T1*EXP(-.946*T1*T1)
      X0=-60.*TR+170.
      XM=1./(.0853+1.5078*(TR-.5952)**1.7336)
```

```
      IF(TR.GT.2.0)WRITE(NW,1110)
      GO TO 400
C
180   Z=.23
      C1=TR/(.003147*EXP(6.0739*TR)-61.12*(TR-1.)*EXP(-19.933*(TR-1.))
     1+1.9036*EXP(-.911*(10.*(TR-.88))**10.)
      C2=EXP(17.671-17.63*TR)+.17-1.1738*EXP(-17.193*ABS(TR-1.05))
      C3=EXP(52.268-49.188*TR)+24.249*EXP(-920.5*(TR-1.1)*(TR-1.1))
      C4=.0032456*EXP(3.2917*TR)-EXP(17.3*TR-30.6)
      C5=1.9772*EXP(-2.1839*TR)
      C6=.000336*EXP(1.61*TR)-.0000054*TR**11.5
      XM=1./(-.023+1.0743*(TR-.4146)**1.8103)
      X0=515.-250.*TR
      IF(TR.GT.1.7)WRITE(NW,1110)
      GO TO 400
C
C
C     SUBCOOLED LIQUID PHASE
C
300   ATR=ALOG(TR)
      APR=ALOG(PR)
      P1=PR-4.2
      T1=TR-.77
      IF(TR.LT.0.5)WRITE(NW,1110)
      GO TO (340,330,320,310),IZC
C
310   Z=.29
      HOHTC=-.09572107*P1+T1*(-9.501235-17.30389*T1-.3195707*P1)
     1+APR*(1.368092+ATR*(4.227096+ATR*3.181639))+9.707447
      GO TO 410
C
320   Z=.27
      P1=P1-0.464
      T1=T1-0.02749
      HOHTC=-.1368774*P1+T1*(-14.56975-7.812724*T1-.1642482*P1)
     1+APR*(1.036851+ATR*(4.463472+ATR*4.525831))+10.86085
      GO TO 410
C
330   Z=.25
      HOHTC=-.1074635*P1+T1*(-15.80132-15.18611*T1-.1476876*P1)
     1+APR*(.7800774+ATR*(3.154058+ATR*2.988533))+12.28618
      GO TO 410
```

```
      HOHTC=-.08644293*P1+T1*(-12.93869+10.81311*T1-.1568094*P1)
     1+APR*(.7466842*ATR*(3.17422+ATR*2.930566))+12.72429
      GO TO 410
C
C
C     SATURATED VAPOUR LINE
C
500   APR=ALOG(PR)
      GO TO(750,700,650,600),IZC
C
600   Z=0.29
      HOHTC=5.4*PR**0.6747/(1.0+1.227*(-APR)**0.503)
      GO TO 410
C
650   Z=0.27
      HOHTC=5.8*PR**0.63163/(1.0+1.229*(-APR)**0.55456)
      GO TO 410
C
700   Z=0.25
      HOHTC=6.5*PR**0.62252/(1.0+0.76218*(-APR)**0.53042)
      GO TO 410
C
750   Z=0.23
      HOHTC=7.0*PR**0.65135/(1.0+0.75727*(-APR)**0.46108)
      GO TO 410
C
C
C     SATURATED LIQUID LINE
C
800   APR=ALOG(PR)
      GO TO (1000,950,900,850),IZC
C
850   Z=0.29
      HOHTC=(5.4+3.6485*(-APR)**0.33464)/(1.0-0.0056742*APR)
      GO TO 410
C
900   Z=0.27
      HOHTC=(5.8+5.19*(-APR)**0.4963)/(1.0-0.1*APR)
      GO TO 410
C
```

```
950   Z=0.25
      HOHTC=(6.5+4.48*(-APR)**0.3952)/(1.0-0.00185*APR)
      GO TO 410
C
1000  Z=0.23
      HOHTC=(7.0+4.5688*(-APR)**0.333)/(1.0+0.004*APR)
      GO TO 410
C
C
400   T1=(ATAN(C3*(1.-PR))/PI+.5)**2
      T1=1.-C2-C4-C5*PR+C2*T1
      T1=T1*EXP(-C1*PR*PR)+C4*PR*(C5+PR*C6)
      HOHTC=XM*PR*(1.-PR/XO)/T1
410   IF(ABS(ZC-Z)-.002)450,450,420
420   IF(KALL-1)430,430,440
430   HOHTC1=HOHTC
      KALL=2
      Z1=Z
      IF(IZC-4)435,433,433
433   IZC=IZC-1
      ZZ=-.02
      GO TO 5
435   IZC=IZC+1
      ZZ=.02
      GO TO 5
440   HOHTC=(HOHTC-HOHTC1)/ZZ *(ZC-Z1)+HOHTC1
450   HOH=HOHTC
      RETURN
1110  FORMAT(/10X,36H ** WARNING FROM ROUTINE ENTHALPY **/
     *10X,48H SYSTEM TEMPERATURE OUTSIDE RANGE OF CORRELATION /
     *10X,41H EXTRAPOLATION USED BUT RESULTS UNCERTAIN /)
      END
      SUBROUTINE TOSI(TRANSFER)
C
      DIMENSION TRANSFER(30)
C
      TRANSFER(1)=TRANSFER(1)*.454
      TRANSFER(2)=(TRANSFER(2)-32.)/1.8
      TRANSFER(3)=TRANSFER(3)*0.0689
      TRANSFER(4)=TRANSFER(4)*2.32
      RETURN
      END
```

```
      DIMENSION TRANSFER(30)
C
      TRANSFER(1)=TRANSFER(1)/.454
      TRANSFER(2)=TRANSFER(2)*1.8+32.
      TRANSFER(3)=TRANSFER(3)/.0689
      TRANSFER(4)=TRANSFER(4)/2.32
      RETURN
      END
      SUBROUTINE TOMASS(TRANSFER)
C
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL
      DIMENSION TRANSFER(30)
C
      IF(NOCOMP.EQ.0)RETURN
      SUM=0.
      DO 10 I=1,NOCOMP
10    SUM=SUM+TRANSFER(NL+I)*PROPRT(I,4)
      TRANSFER(1)=TRANSFER(1)*SUM
      DO 20 I=1,NOCOMP
20    TRANSFER(NL+I)=TRANSFER(NL+I)*PROPRT(I, 4)/SUM
      RETURN
      END
      SUBROUTINE TOMOLE(TRANSFER)
C
      DIMENSION TRANSFER(30)
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL
C
      IF(NOCOMP.EQ.0)RETURN
      SUM=0.
      DO 10 I=1,NOCOMP
      SUM=SUM+TRANSFER(NL+I)/PROPRT(I,4)
10    CONTINUE
      DO 20 I=1,NOCOMP
20    TRANSFER(NL+I)=TRANSFER(NL+I)/PROPRT(I, 4)/SUM
      SUM=0.
      DO 30 I=1,NOCOMP
30    SUM=SUM+TRANSFER(NL+I)*PROPRT(I,4)
      TRANSFER(1)=TRANSFER(1)/SUM
```

```
      RETURN
      END
XXX   SUBROUTINE KVALUE(IRL,TRANSFER,EQCON)
C
      DIMENSION A(10,3),TRANSFER(30),EQCON(10),X(10),Y(10),VLOG(10)
      DIMENSION PR(10)
      COMMON /RK/BIGA(10),BIGB(10),AABAR,BBBAR,ABAR
      COMMON /PROPS/ PROPRT(10,25),USERPPT(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL,NW
C
C
      DATA R/1.9872/,A/1.96718,1.02972,-.054009,.0005288,0.0,0.008585,
     14*0.0,2.4384
     1,-2.2455,-0.34084,0.00212,-0.00223,0.10486,-0.03691,3*0.0,5.75748,
     2-3.01761,-4.985,2.02299,0.0,.08427,0.26667,-0.31138,-0.02655,
     30.02883/
C
      IF(NOCOMP.EQ.1)GO TO 240
      IF(IUNITS)1,1,2
1     FACT=1.8
      TADD=460.
      P2000=2000.
      GO TO 3
2     TADD=273.15
      FACT=1.0
      P2000=140.
3     Z1=.9
      IF(TRANSFER(2).EQ.0.0)TRANSFER(2)=TRANSFER(NSLMAX+2)
      IF(TRANSFER(3).EQ.0.0)TRANSFER(3)=TRANSFER(NSLMAX+3)
      TABS=TRANSFER(2)+TADD
      JJ=0
      DO 10 I=1,NOCOMP
      X(I)=TRANSFER(NL+I)
      Y(I)=TRANSFER(NSLMAX+NL+I)
      IF(Y(I).GT.0.0.AND.X(I).EQ.0.0)X(I)=Y(I)
      IF(X(I).GT.0.0.AND.Y(I).EQ.0.0)Y(I)=X(I)
      IF(Y(I).GT.0.005)JJ=JJ+1
10    CONTINUE
      IF(JJ.LE.1)GO TO 240
      SUM=0.
      KALL=0
      DO 110 I=1,NOCOMP
```

```
      IF(P.GT.P2000.OR.PR(I).GT.0.8)WRITE(NW,310)PROPRT(I,1)
     *,PROPRT(I,2),PR(I)
      IF(TR.LT.0.5.OR.TR.GT.1.3)WRITE(NW,320)PROPRT(I,1),PROPRT(I,2),TR
      IF(PROPRT(I,3)-2.)40,50,60
40    N=1
      GO TO 70
50    N=2
      GO TO 70
60    N=3
70    VOLOG=A(1,N)+A(2,N)/TR+TR*(A(3,N)+TR*(A(4,N)+TR*A(5,N)))
     1+PR(I)*(A(6,N)+TR*(A(7,N)+TR*A(8,N))+PR(I)*(A(9,N)+A(10,N)*TR))
      V1LOG=-4.2389+TR*(8.65808-3.15224*TR*TR)-1.2206/TR-0.025
     **(PR(I)-.6)
110   VLOG(I)=VOLOG+PROPRT(I,9)*V1LOG
      SUM1=0.
      CALL RKDATA(Y,TRANSFER(2))
4     C1=AABAR*AABAR/BBBAR
      C2=BBBAR*TRANSFER(3)
      C3=C2*C1
      C4=C2*C2
5     J=0
80    Z2=Z1*Z1
      Z3=Z2*Z1
      C5=C3-C4-C2
      F=Z3-Z2+C5*Z1-C1*C4
      FPR=3.0*Z2-2.0*Z1+C5
      Z=Z1-F/FPR
      J=J+1
      IF(ABS(Z-Z1)-ABS(.001*Z))111,111,90
90    IF(J-30)100,100,111
100   Z1=Z
      GO TO 80
111   ALOGCZ=ALOG(1.0+C2/Z)
      IF(KALL)25,25,35
 20    IF(IRL)25,21,25
21    CALL RKDATA(Y,TRANSFER(2))
      KALL=1
      GO TO 4
25    D1=0.
      D2=0.
```

```
      DO 30 I=1,NOCOMP
      DBAR=X(I)*PROPRT(I,11)
      D2=D2+DBAR
30    D1=D1+DBAR*PROPRT(I,10)
      DBAR=D1/D2
35    DO 120 I=1,NOCOMP
      BIGBR=BIGB(I)/BBBAR
      GAMALN=(PROPRT(I,11)*FACT*(PROPRT(I,10)-DBAR)**2/R/TABS)
      PHILN=(Z-1.0)*BIGBR-ALOG(Z-C2)-C1*(2.0*BIGA(I)/AABAR-BIGBR)*
     *ALOGCZ
      EQA=GAMALN-PHILN
      EQB=EXP(EQA)
      EQC=EXP10(VLOG(I))
      EQD=EQC*EQB
120   EQCON(I)=EQD/PR(I)
      SUM1=SUM
      SUM=0.
      DO 150 I=1,NOCOMP
      IF(IRL)140,130,140
130   X(I)=Y(I)/EQCON(I)
      SUM=SUM+X(I)
      GO TO 150
140   Y(I)=X(I)*EQCON(I)
      SUM=SUM+Y(I)
150   CONTINUE
      IF(ABS(SUM-SUM1)-0.0005)200,200,160
160   DO 190 I=1,NOCOMP
      IF(IRL)180,170,180
170   X(I)=X(I)/SUM
      GO TO 190
180   Y(I)=Y(I)/SUM
190   CONTINUE
      GO TO 20
200   DO 210 I=1,NOCOMP
      TRANSFER(NL+I)=Y(I)
210   TRANSFER(NSLMAX+NL+I)=X(I)
      TRANSFER(NSLMAX+2)=TRANSFER(2)
      TRANSFER(NSLMAX+3)=TRANSFER(3)
      TRANSFER(NSLMAX+5)=0.0
      TRANSFER(5)=1.0
230   RETURN
240   DO 250 I=1,NOCOMP
```

```
      TRANSFER(NSLMAX+3)=TRANSFER(3)
      TRANSFER(5)=1.0
      TRANSFER(NSLMAX+5)=0.0
      RETURN
310   FORMAT(/10X,34H ** WARNING FROM ROUTINE KVALUE ** /
     *10X,35H THE REDUCED PRESSURE OF COMPONENT ,2A8, 5H    (,F5.2,2H )/
     *10X,33H IS OUTSIDE THE RECOMMENDED RANGE /)
320   FORMAT(/10X,34H ** WARNING FROM ROUTINE KVALUE ** /
     *10X,38H THE REDUCED TEMPERATURE OF COMPONENT ,2A8,2H (,F5.2,2H ) /
     *10X,33H IS OUTSIDE THE RECOMMENDED RANGE /)
      END
      SUBROUTINE DEWBUB(IRL,TRANSFER,TF)
C
      DIMENSION X(10),Y(10),EQCON(10),TRANSFER(30),SAVE(15)
      COMMON /PROPS/ PROPRT(10,25),USERPPT(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL,NW
C
      IMAX=1
      PCAV=0.0
      TCAV=0.0
      IF(NOCOMP.LE.1)GO TO 220
      DO 5 I=1,NSLMAX
5     SAVE(I)=TRANSFER(I)
      JJ=0
      TIN=TRANSFER(2)
      TMAX=-1000.
      TMIN=1000.
      DO 30 I=1,NOCOMP
      IF(TRANSFER(NL+I)-.001)30,30,10
10    T=PROPRT(I,8)
      JJ=JJ+1
      IF(TMAX-T)15,20,20
15    IMAX=I
      TMAX=T
20    IF(TMIN-T)26,26,25
25    IMIN=I
      TMIN=T
26    PCAV=PCAV+PROPRT(I,5)*TRANSFER(NL+I)
      TCAV=TCAV+PROPRT(I,6)*TRANSFER(NL+I)
30    CONTINUE
```

```
      TCAV=TCAV-459.7/(1.0+0.8*IUNITS)
      IF(JJ.LE.1)GO TO 220
      IF(TRANSFER(3).GT.PCAV)GO TO 250
      PRES=TRANSFER(3)
      TMIN=BOILPT(IMIN,PRES)
      TMAX=BOILPT(IMAX,PRES)
      T1=(TMIN+TMAX)/2.
      T1=T1+10.
      T2=T1-20.
      TRANSFER(2)=T1
      DO 40 I=1,NOCOMP
      X(I)=TRANSFER(NL+I)
      Y(I)=X(I)
40    TRANSFER(NSLMAX+NL+I)=X(I)
      CALL KVALUE(IRL,TRANSFER,EQCON)
      SUM1=0.
      DO 70 I=1,NOCOMP
      IF(IRL)50,50,60
50    X(I)=Y(I)/EQCON(I)
      SUM1=SUM1+X(I)
      GO TO 70
60    Y(I)=X(I)*EQCON(I)
      SUM1=SUM1+Y(I)
70    CONTINUE
      TRANSFER(2)=T2
75    DO 80 I=1,NOCOMP
      IF(IRL)77,77,78
77    X(I)=X(I)/SUM1
      GO TO 79
78    Y(I)=Y(I)/SUM1
79    TRANSFER(NL+I)=Y(I)
80    TRANSFER(NSLMAX+NL+I)=X(I)
      CALL KVALUE(IRL,TRANSFER,EQCON)
85    SUM2=0.
      DO 100 I=1,NOCOMP
      IF(IRL)90,90,95
90    X(I)=Y(I)/EQCON(I)
      SUM2=SUM2+X(I)
      GO TO 100
95    Y(I)=X(I)*EQCON(I)
      SUM2=SUM2+Y(I)
100   CONTINUE
```

```
      TRANSFER(2)=TF
150   T1=T2
      T2=TF
      SUM1=SUM2
      GO TO 75
200   DO 205 I=1,NSLMAX
205   TRANSFER(I)=SAVE(I)
      RETURN
220   TF=BOILPT(IMAX,TRANSFER(3))
      RETURN
250   WRITE(NW,260)TRANSFER(3),PCAV,TCAV
260   FORMAT(/10X,34H ** WARNING FROM ROUTINE DEWBUB ** /
     *10X,18H SYSTEM PRESSURE (,F10.3,15H ) HIGHER THAN /
     *10X,28H AVERAGE CRITICAL PRESSURE (,F10.3,2H ) /
     *10X,47H DEW/BUBBLE POINT SET EQUAL TO AVERAGE CRITICAL /
     *10X,12H TEMPERATURE , F10.3/)
      TF=TCAV
      RETURN
      END
      SUBROUTINE FLASH(TRANSFER, PRES)
C
      DIMENSION Z(10),EQCON(10),TRANSFER(30)
      COMMON /PROPS/ PROPRT(10,25),USERPPT(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL,NW
C
      TRANSFER(3)=PRES
      JJ=0
      RL=.2
      II=0
      IRL=1
      PCAV=0.0
      DO 20 I=1,NOCOMP
      TRANSFER(NSLMAX+NL+I)=TRANSFER(NL+I)
      IF(TRANSFER(NL+I).LT.0.005)GO TO 20
       JJ=JJ+1
      PCAV=PCAV+TRANSFER(NL+I)*PROPRT(I,5)
      IJ=I
20    Z(I)=TRANSFER(NL+I)
      I=IJ
      IF(NOCOMP.LE.1)GO TO 270
```

```
      IF(JJ.LE.1)GO TO 270
      IF(PCAV.LT.TRANSFER(3))GO TO 320
      CALL KVALUE(IRL,TRANSFER,EQCON)
      SUMX=0.
      DO 30 I=1,NOCOMP
      TRANSFER(NSLMAX+NL+I)=Z(I)/(RL+(1.-RL)*EQCON(I))
30    SUMX=SUMX+TRANSFER(NSLMAX+NL+I)
40    SUMY=0.
      DO 50 I=1,NOCOMP
      TRANSFER(NSLMAX+NL+I)=TRANSFER(NSLMAX+NL+I)/SUMX
      TRANSFER(NL+I)=TRANSFER(NSLMAX+NL+I)*EQCON(I)
50    SUMY=SUMY+TRANSFER(NL+I)
      DO 60 I=1,NOCOMP
60    TRANSFER(NL+I)=TRANSFER(NL+I)/SUMY
      CALL KVALUE(IRL,TRANSFER,EQCON)
      FRL=0.
      FPRL=0.
      SUMX=0.
      DO 70 I=1,NOCOMP
      TRANSFER(NSLMAX+NL+I)=Z(I)/(RL+(1.-RL)*EQCON(I))
      SUMX=SUMX+TRANSFER(NSLMAX+NL+I)
      TRANSFER(NL+I)=TRANSFER(NSLMAX+NL+I)*EQCON(I)
      FRL=FRL+TRANSFER(NSLMAX+NL+I)
70    FPRL=FPRL-TRANSFER(NSLMAX+NL+I)*(1.-EQCON(I))/(RL+(1.-RL)*EQCON(I)
     1)
      FRL=FRL-1.
      II=II+1
      IF(II-30)80,80,150
80    RL1=RL-FRL/FPRL
90    IF(ABS(FRL)-.1E-04)120,120,100
100   IF(ABS(RL1)-.1E08)110,110,170
110   RL=RL1
      IF(FRL)190,120,230
120   IF(ABS(RL-1.)-.1E-03)170,170,150
130   IF(ABS(RL)-.1E-03)150,150,260
150   DO 160 I=1,NOCOMP
      TRANSFER(NL+I)=Z(I)
160   TRANSFER(NSLMAX+NL+I)=0.
      RL=0.
      GO TO 260
170   DO 180 I=1,NOCOMP
      TRANSFER(NL+I)=0.
```

```
      GO TO 260
190   IF(FPRL)210,150,150
210   IF(RL1)150,150,40
230   IF(FPRL)40,240,240
240   WRITE(NW,250)
      STOP
250   FORMAT(//10X,'ERROR IN LINE 230  OF FLASH'//)
260   TRANSFER(5)=1.0
      TRANSFER(NSLMAX+2)=TRANSFER(2)
      TRANSFER(NSLMAX+3)=PRES
      TRANSFER(NSLMAX+5)=0.0
      FLOW=TRANSFER(1)
      IF(RL.GT.0.995)RL=1.0
      IF(RL.LT.0.005)RL=0.0
      TRANSFER(NSLMAX+1)=FLOW*RL
      TRANSFER(1)=FLOW-TRANSFER(NSLMAX+1)
      RETURN
270   CALL DEWBUB(1,TRANSFER,T)
      IF(TRANSFER(2)-T)280,300,290
280   DO 285 I=1,NOCOMP
      TRANSFER(NSLMAX+NL+I)=TRANSFER(NL+I)
285   TRANSFER(NL+I)=0.
      RL=1.
      GO TO 260
290   DO 295 I=1,NOCOMP
295   TRANSFER(NSLMAX+NL+I)=0.0
      RL=0.0
      GO TO 260
300   DO 310 I=1,NOCOMP
310   TRANSFER(NSLMAX+NL+I)=TRANSFER(NL+I)
      VF=PUREVF(I,TRANSFER)
      RL=1.0-VF
      GO TO 260
320   WRITE(NW,330)TRANSFER(3),PCAV
330   FORMAT(/10X,33H ** WARNING FROM ROUTINE FLASH **  /
     *10X,18H SYSTEM PRESSURE (,F10.3,14H ) HIGHER THAN /
     *10X,27H SYSTEM CRITICAL PRESSURE (,F10.3,2H ) /
     *10X,22H SYSTEM ASSUMED VAPOUR /)
      GO TO 150
      END
```

```
      SUBROUTINE ENTHALPY(TRANSFER)
C
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL,NW
      DIMENSION TRANSFER(30),SAVE(15)
C
      VF=TRANSFER(5)
      HOH=0.0
      IF(IUNITS.GT.0)GO TO 2
      TADD=460.
      R=10.7335
      GO TO 5
2     R=0.084778
      TADD=273.15
5     DO 10 I=1,NSLMAX
10    SAVE(I)=TRANSFER(I)
      KALL=0
      IF(TRANSFER(5))40,60,30
30    IF(TRANSFER(5)-1.)135,90,33
33    IF(TRANSFER(5).GT.2.0.AND.TRANSFER(5).LT.3.0)GO TO 130
35    IF(TRANSFER(5)-2.0)90,36,37
36    ISTATE=2
      GO TO 100
37    ISTATE=3
      GO TO 100
40    CALL DEWBUB(1,TRANSFER,BP)
      IF(TRANSFER(2)-BP)50,50,70
50    TRANSFER(5)=0.
60    ISTATE=0
      SAVE(5)=0.0
      GO TO 100
70    CALL DEWBUB(0,TRANSFER,DP)
      IF(TRANSFER(2)-DP)130,80,80
80    TRANSFER(5)=1.
      SAVE(5)=1.0
90    ISTATE=1
100   PC=0.0
      TC=0.0
      ZC=0.0
105   DO 120 I=1,NOCOMP
      IF(TRANSFER(NL+I)-0.001)120,120,110
110   PC=PC+PROPRT(I,5)*TRANSFER(NL+I)
```

```
120   CONTINUE
      IF(ABS(PC).LT.0.01.OR.ABS(TC).LT.0.01)GO TO 140
      TR=(TRANSFER(2)+TADD)/TC
      PR=TRANSFER(3)/PC
      CALL HDATA(ISTATE,TR,PR,ZC,HOH,NW)
      GO TO 140
130   PRES=TRANSFER(3)
      CALL FLASH(TRANSFER,PRES)
      VF=TRANSFER(1)/SAVE(1)
135   SAVE(5)=VF
      ISTATE=1
      KALL=1
      GO TO 100
140   IF(KALL-1)145,150,170
145   HOH=HOH*TC
      GO TO 180
150   KALL=2
      DO 160  I=1,NSLMAX
160   TRANSFER(I)=TRANSFER(NSLMAX+I)
      ISTATE=0
      HD1=HOH*TC
      GO TO 100
170   HOH=HD1*VF+HOH*(1.-VF)*TC
180   HO=0.
      T=SAVE(2)
      IF(IUNITS.EQ.0)GO TO 185
      HOH=HOH*2.325
      T=T*1.8+32.
185   DO 190 I=1,NOCOMP
190   HO=HO+SAVE(NL+I)*PROPRT(I, 4)*(PROPRT(I,14)+T*(PROPRT(I,15)+
     1T*(PROPRT(I,16)+T*(PROPRT(I,17)+T*(PROPRT(I,18)+T*PROPRT(I,19))))
     2)
      SAVE(4)=HO-HOH
      DO 200 I=1,NSLMAX
200   TRANSFER(I)=SAVE(I)
      RETURN
      END
      SUBROUTINE HSAT(TRANSFER)
C
      DIMENSION TRANSFER(30)
```

```
C
      IF(TRANSFER(5).GT.1.9)GO TO 10
      TRANSFER(5)=TRANSFER(5)+2.
10    CALL ENTHALPY(TRANSFER)
      TRANSFER(5)=TRANSFER(5)-2.
      RETURN
      END
      SUBROUTINE TEMP(TRANSFER)
C
      DIMENSION TRANSFER(30)
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL
C
      HIN=TRANSFER(4)
      VF=TRANSFER(5)
      IF(VF)5,25,5
5     CALL DEWBUB(0,TRANSFER,DP)
      TRANSFER(2)=DP
      TRANSFER(5)=1.
      CALL ENTHALPY(TRANSFER)
      HDP=TRANSFER(4)
      K=0
      IF(ABS(HDP-HIN)-2.0)40,40,10
10    IF(HDP-HIN)60,40,20
20    IF(VF-.905)25,40,40
25    CALL DEWBUB(1,TRANSFER,BP)
      TRANSFER(2)=BP
      TRANSFER(5)=0.
      CALL ENTHALPY(TRANSFER)
      HBP=TRANSFER(4)
      IF(ABS(HBP-HIN)-2.0)50,50,30
30    IF(HBP-HIN)155,50,140
40    TRANSFER(2)=DP
      TRANSFER(4)=HIN
      RETURN
50    TRANSFER(2)=BP
      TRANSFER(4)=HIN
      RETURN
60    T1=DP
      T2=T1+40.
      H1=HDP
      TRANSFER(2)=T2
```

```
      CALL ENTHALPY(TRANSFER)
      H2=TRANSFER(4)
70    T=T1-(T1-T2)*(H1-HIN)/(H1-H2)
      IF(KALL-1)80,90,100
80    TRANSFER(5)=0.
      GO TO 110
90    TRANSFER(5)=2.5
      GO TO 110
100   TRANSFER(5)=1.
110      TRANSFER(2)=T
      CALL ENTHALPY(TRANSFER)
      K=K+1
      H=TRANSFER(4)
      IF(K-15)120,120,160
120   IF(ABS(H-HIN)-5.0)160,160,130
130   T2=T1
      T1=T
      H2=H1
      H1=H
      GO TO 70
140   T1=BP
      T2=T1-40.
      H1=HBP
      TRANSFER(2)=T2
      TRANSFER(5)=0.
      KALL=0
      CALL ENTHALPY(TRANSFER)
      H2=TRANSFER(4)
      GO TO 70
150   IF(ABS(VF)-0.005)50,50,155
155   IF(ABS(DP-BP).GT.0.5)GO TO 158
      TRANSFER(2)=(BP+DP)/2
      J=0
      DO 156 I=1,NOCOMP
      IF(TRANSFER(NL+I).LT.0.005)GO TO 156
      IJ=I
      J=J+1
156   CONTINUE
      I=IJ
      TRANSFER(5)=PUREVF(I,TRANSFER)
```

```
      RETURN
158   T1=DP
      T2=BP
      H1=HDP
      H2=HBP
      KALL =1
      GO TO 70
160   TRANSFER(2)=T
      TRANSFER(4)=HIN
      RETURN
      END
      FINISH
```

Program MODELSUBS

```
      LIST(LP)
      SEGMENTS(FXXX)
      INPUT 1 = CR0
      OUTPUT 2 = LP0
      COMPRESS INTEGER AND LOGICAL
      TRACE 2,500
      END
C
C
      SUBROUTINE COMPRESR
C
      DIMENSION TRANSFER(30)
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
     *,NW
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS
C
C
C *** MODEL OF A POLYTROPIC GAS OMPRESSOR.
C *** THE RATIO OF (T OUT/T IN) IS SET EQUAL TO THE RATIO
C *** OF (P OUT/P IN) TO THE POWER ((K-1)/K), WHERE
C *** K IS THE RATIO OF SPECIFIC HEATS (CP/CV).
C
      IF(MODE.GT.0)GO TO 155
      DO 10 I=1,NSLMAX
      STRMO(1,I)=STRMI(1,I)
10    TRANSFER(I)=STRMI(1,I)
      IF(PARAM(1).EQ.0.0)GO TO 180
      STRMO(1,3)=PARAM(1)
      TRANSFER(3)=PARAM(1)
      IF(STRMI(1,5)-1.)20,40,40
20    WRITE(NW,30)
30    FORMAT(/10X,34H OUTPUT FROM COMPRESSOR IS NOT GAS/)
      TRANSFER(5)=-1.
      CALL TEMP(TRANSFER)
      STRMO(1,2)=TRANSFER(2)
      STRMO(1,5)=TRANSFER(5)
      RETURN
40    CALL DEWBUB(0,TRANSFER,DPOUT)
      EFFS=PARAM(2)
      EFFM=PARAM(3)
```

```
50      EFFS=.75
        PARAM(2)=.75
60      IF(EFFM)70,70,80
70      EFFM=.95
        PARAM(3)=.95
80      INDEX=0
        IF(CPCV)90,90,110
C *** IF THE RATIO OF SPECIFIC HEATS IS NOT GIVEN, CALCULATE IT
C *** USING ROUTINE  SPHEAT.
90      INDEX=1
        TRANSFER(2)=STRMI(1,2)-5.
100     TRANSFER(5)=1.
        CALL ENTHALPY(TRANSFER)
        H1=TRANSFER(4)
        TRANSFER(2)=TRANSFER(2)+10.
        CALL ENTHALPY(TRANSFER)
        H2=TRANSFER(4)
        CP=(H2-H1)/10.
        CV=CP-R
        CPCV=CP/CV
110     POWER=(CPCV-1.)/CPCV
        TIN=STRMI(1,2)
        TADD=460.
        IF(IUNITS.EQ.1)TADD=273.15
        TIN=TIN+TADD
        T2=TIN*(STRMO(1,3)/STRMI(1,3))**POWER
        TOUT=(T2-TADD-STRMI(1,2))/EFFS+STRMI(1,2)
        IF(TOUT-DPOUT)20,120,120
120     IF(INDEX-1)150,130,140
130     TRANSFER(2)=TOUT-5.
        TOUT1=TOUT
        INDEX=2
        GO TO 100
140     IF(ABS(TOUT-TOUT1)-0.5)150,150,130
150     TRANSFER(2)=TOUT
        TRANSFER(5)=1.
        CALL ENTHALPY(TRANSFER)
        STRMO(1,2)=TRANSFER(2)
        STRMO(1,4)=TRANSFER(4)
        STRMO(1,5)=1.
        RETURN
```

```
155   ENERGY=(STRMO(1,4)-STRMI(1,4))*STRMI(1,1)/PARAM(3)
      WRITE(NW,160)ENERGY
160   FORMAT(/10X,32H POWER REQUIRED BY COMPRESSOR = ,1E15.7,
     *' ENTHALPY UNITS')
170   RETURN
180   WRITE(NW,190)
190   FORMAT(//10X,36H ** WARNING FROM ROUTINE COMPRESR ** /
     *10X,47H OUTLET PRESSURE NOT SET, SIMULATION ABANDONNED //)
      PAUSE ER
      END
C
C
      FUNCTION DEG(KDEG,IUNITS,T)
C
C *** THIS FUNCTION IS USED BY THE DISCOL MODEL WHEN THE USER
C *** SUPPLIES HIS OWN K-VALUE CORRELATIONS USING WHICHEVER
C *** TEMPERATURE UNITS HE WISHES.  IT MERELY RECONCILES THE
C *** UNITS OF THE SIMULATION WITH THOSE OF THE K-VALUE
C *** CORRELATIONS.
C
C     KDEG = 1  IF DEG RETURNED IN DEGREES F
C            2  IF DEG RETURNED IN DEGREES C
C            3  IF DEG RETURNED IN DEGREES R
C            4  IF DEG RETURNED IN DEGREES K
C
C
      IF(IUNITS.EQ.1)GO TO 5
      GO TO (1,2,3,4),KDEG
1     DEG=T
      RETURN
2     DEG=(T-32.0)/1.8
      RETURN
3     DEG=T+460.0
      RETURN
4     DEG=(T-32.0)/1.8+273.15
      RETURN
5     GO TO (6,1,7,8),KDEG
6     DEG=T*1.8+32.0
      RETURN
7     DEG=T*1.8+492.0
```

```
      END
C
C
      SUBROUTINE EVALUE(L,M,T,EQCON)
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS
      DIMENSION EQCON(10),DIV(6)
      DATA DIV/.1E03,.1E05,.1E08,.1E11,.1E13,.1E15/
C
C
C *** VAPOUR-LIQUID EQUILIBRIUM CONSTANTS (KVALUES).
C *** THIS ROUTINE GENERATES K-VALUES FROM USER PROVIDED POLYNOMIALS
C *** OF K VS. TEMPERATURE.
C *** THE POLYNOMIALS ARE OF ORDER L-1 AND THE COEFFICIENTS GIVEN
C *** IN THE UNIT PARAMETER ARRAY STARTING IN POSITION M.
C
      K=M-L
      DO 20 I=1,NOCOMP
      K=K+L
      EQCON(I)=PARAM(K)
      TN=1.0
      DO 10 J=1,L-1
      TN=TN*T
10    EQCON(I)=EQCON(I)+TN*PARAM(K+J)/DIV(J)
20    CONTINUE
      RETURN
      END
C
C
      SUBROUTINE DISCOL
C
      DIMENSION XX(10),YY(10),TRANSFER(30)
      DIMENSION EQCON(10,40),XD1(10)
      DIMENSION HL(40),HV(40),V(40),RL(40),HF(40)
      DIMENSION T(40),TX3(40),TY2(40),TY3(40)
      DIMENSION F(40),W(40),U(40),Q(40)
      DIMENSION A(40),B(40),C(40),D(40),P(40),QQ(40)
      DIMENSION X(10,40),Y(10,40),Z(10,40)
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
     *,NW
```

```
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS
C
C *** MODEL OF A GENERAL MULTICOMPONENT, TRAY-TO-TRAY
C *** DISTILLATION COLUMN, USING THE WANG AND HENKE METHOD.
C *** (HYDROCARBON PROCESSING, VOL. 45, NO. 8, PAGE 155, 1966)
C
      IF(MODE.GT.0)GO TO 570
C
C *** SETTING THE SIMULATION PARAMETERS.
C
6     N=PARAM(1)
      JFD=PARAM(2)
      DD=PARAM(3)
      BB=STRMI(1,1)-DD
      V(1)=PARAM(4)
      R=PARAM(5)
      PRES=PARAM(6)
      IF(PRES.LE.STRMI(1,3))GO TO 8
      WRITE(NW,7)
7     FORMAT(/10X,34H ** WARNING FROM ROUTINE DISCOL ** /
     *10X,54H COLUMN PRESSURE SET HIGHER THAN INPUT STREAM PRESSURE /
     *10X,50H SETTING IGNORED AND STREAM PRESSURE USED INSTEAD /)
      PRES=STRMI(1,3)
8     KEYTOP=PARAM(7)
      KEYBOT=PARAM(8)
      IF(PRES.EQ.0.0)PRES=STRMI(1,3)
      PCAV=0.0
      DO 9 I=1,NOCOMP
9     PCAV=PCAV+STRMI(1,NL+I)*PROPRT(I,5)
      IF(PRES.LT.PCAV)GO TO 12
      WRITE(NW,10)
10    FORMAT(//10X,34H ** WARNING FROM ROUTINE DISCOL ** /
     *10X,55H COLUMN PRESSURE GREATER THAN AVERAGE CRITICAL PRESSURE  /
     *10X,18H ALL OUTPUT IS GAS /)
      DO 11 I=1,NSLMAX
      STRMO(1,I)=STRMI(2,I)
11    STRMO(2,I)=0.0
      Q(1)=0.0
      Q(N)=0.0
      GO TO 480
```

```
      KDEG=PARAM(11)
      ISTART=12
      XTOP1=.95
      XBOT2=.95
      XTOP2=.05
      XBOT1=.05
C
C *** ZEROING THE ARRAYS AND SETTING THE FEED TRAY VALUES.
C
      DO 15 I=1,N
      Q(I)=0.
      U(I)=0.
      W(I)=0.
      F(I)=0.
      HF(I)=0.
      DO 15 J=1,NOCOMP
      Z(J,I)=0.
      TRANSFER(NL+J)=0.
15    CONTINUE
      KB=0
      F(JFD)=STRMI(1,1)
      HF(JFD)=STRMI(1,4)
      DO 20 J=1,NOCOMP
20    Z(J,JFD)=STRMI(1,NL+J)
      U(1)=DD-V(1)
      XN=N
      TRANSFER(NL+KEYTOP)=XTOP1
      TRANSFER(NL+KEYBOT)=XTOP2
      TRANSFER(3)=PRES
      CALL DEWBUB(0,TRANSFER,T(1))
      TRANSFER(NL+KEYTOP)=XBOT1
      TRANSFER(NL+KEYBOT)=XBOT2
      CALL DEWBUB(1,TRANSFER,T(N))
C
C *** INITIAL  X  PROFILE
C
      M=NOCOMP-KEYTOP
      XV=(XBOT1-XTOP1)/(XN-1.)
      DO 30 I=1,KEYTOP
      DO 30 J=1,N
      XJ=J-1
```

```
      X(I,J)=XTOP1+XV*XJ
30    Y(I,J)=X(I,J)
      XV=(XBOT2-XTOP2)/(XN-1.)
      DO 40 I=1,M
      DO 40 J=1,N
      XJ=J-1
      II=I+KEYTOP
      X(II,J)=XTOP2+XV*XJ
40    Y(II,J)=X(II,J)
C
C *** INITIAL TEMPERATURE AND VAPOUR FLOW RATE  PROFILES
C
      V(2)=(R+1.)*DD
      V(N)=DD*R
      XV=(V(N)-V(2))/(XN-2.)
      XT=(T(N)-T(1))/(XN-1.)
      DO 90 J=2,N
      XJ1=J-2
      XJ=J-1
      V(J)=V(2)+XV*XJ1
      T(J)=T(1)+XT*XJ
      TY2(J)=T(J)
90    CONTINUE
      TY2(1)=T(1)
C
C *** INITIAL EQUILIBRIUM CONSTANTS
C
      DO 110 J=1,N
      IF(NTERMS.EQ.0)GO TO 105
      XTT=DEG(KDEG,IUNITS,T(J))
      CALL EVALUE(NTERMS,ISTART,XTT,XD1)
      GO TO 108
105   DO 100 I=1,NOCOMP
      TRANSFER(NL+I)=Y(I,J)
100   TRANSFER(NSLMAX+NL+I)=X(I,J)
      TRANSFER(2)=T(J)
      IRL=1
      CALL KVALUE(IRL,TRANSFER,XD1)
108   DO 110 I=1,NOCOMP
      EQCON(I,J)=XD1(I)
```

```
C *** GENERAL TRIDIAGONAL MATRIX FOR DISTILLATION
C
120     SUM=0.
        RL(1)=V(2)-DD
        DO 130 KA=2,N-1
        SUM=SUM+F(KA)-W(KA)-U(KA)
130     RL(KA)=V(KA+1)+SUM-DD
        RL(N)=BB
        D(1)=0.
        D(N)=0.
        A(N)=V(N)+BB
        DO 140 J=2,N-1
140     A(J)=RL(J-1)
        DO 190 I=1,NOCOMP
        B(1)=-(V(1)*EQCON(I,1)+U(1)+RL(1) )
        C(1)=V(2)*EQCON(I,2)
        B(N)=-(V(N)*EQCON(I,N)+BB)
        DO 150 J=2,N-1
        B(J)=-((V(J)+W(J))*EQCON(I,J)+(RL(J)+U(J)))
        C(J)=V(J+1)*EQCON(I,J+1)
        D(J)=-F(J)*Z(I,J)
150     CONTINUE
        P(1)=C(1)/B(1)
        QQ(1)=D(1)/B(1)
        DO 160 J=2,N-1
160     P(J)=C(J)/(B(J)-A(J)*P(J-1))
        DO 170 J=2,N
170     QQ(J)=(D(J)-A(J)*QQ(J-1))/(B(J)-A(J)*P(J-1))
        X(I,N)=QQ(N)
        DO 180 JK=1,N-1
        J=N-JK
180     X(I,J)=QQ(J)-P(J)*X(I,J+1)
190     CONTINUE
        DO 220 J=1,N
        SSUM=0.0
        DO 200 I=1,NOCOMP
200     SSUM=X(I,J)+SSUM
        DO 210 I=1,NOCOMP
210     X(I,J)=X(I,J)/SSUM
220     CONTINUE
C
```

```
C *** COMPARE CLOSENESS OF FIT FOR TWO DIFFERENT TEMPERATURES
C
      LMAX=30
      DO 380 J=1,N
      Z1=T(J)-10.
      L=0
      SUM1=0.0
      IF(NTERMS.EQ.0)GO TO 225
      XTT=DEG(KDEG,IUNITS,Z1)
      CALL EVALUE(NTERMS,ISTART,XTT,XD1)
      GO TO 235
225   DO 230 I=1,NOCOMP
      TRANSFER(NL+I)=Y(I,J)
230   TRANSFER(NSLMAX+NL+I)=X(I,J)
      TRANSFER(2)=Z1
      IRL=-1
      CALL KVALUE(IRL,TRANSFER,XD1)
235   DO 240 I=1,NOCOMP
      EQCON(I,J)=XD1(I)
240   SUM1=SUM1+EQCON(I,J)*X(I,J)
      FZ1=SUM1-1.
      Z2=T(J)+10.0
      SUM1=0.
      IF(NTERMS.EQ.0)GO TO 250
      XTT=DEG(KDEG,IUNITS,Z2)
      CALL EVALUE(NTERMS,ISTART,XTT,XD1)
      GO TO 265
250   DO 260 I=1,NOCOMP
260   TRANSFER(NSLMAX+NL+I)=X(I,J)
      TRANSFER(2)=Z2
      IRL=1
      CALL KVALUE(IRL,TRANSFER,XD1)
265   DO 270 I=1,NOCOMP
      EQCON(I,J)=XD1(I)
270   SUM1=SUM1+EQCON(I,J)*X(I,J)
      FZ2=SUM1-1.
280   ZZ=Z2-FZ2*(Z2-Z1)/(FZ2-FZ1)
290   L=L+1
      SUM1=0.0
      IF(NTERMS.EQ.0)GO TO 295
```

```
295     DO 300 I=1,NOCOMP
300     TRANSFER(NSLMAX+NL+I)=X(I,J)
        TRANSFER(2)=ZZ
        IRL=1
        CALL KVALUE(IRL,TRANSFER,XD1)
305     DO 310 I=1,NOCOMP
        EQCON(I,J)=XD1(I)
310     SUM1=SUM1+EQCON(I,J)*X(I,J)
        FZZ=SUM1-1.0
320     IF(ABS(FZZ).LT.0.00001)GO TO 380
        IF(L-LMAX)340,330,330
330     WRITE(NW,1000)ZZ
        GO TO 380
340     FZ1=FZ2
        FZ2=FZZ
        Z1=Z2
        Z2=ZZ
        GO TO 280
380     T(J)=ZZ
        SUM =0.
        DO 390 J=1,N
390     SUM=SUM+(T(J)-TY3(J))**2
        DO 400 J=1,N
        DO 400 I=1,NOCOMP
400     Y(I,J)=X(I,J)*EQCON(I,J)
        KB=KB+1
        ET=FLOAT(N)/50.
C       WRITE(NW,405)KB,SUM
405     FORMAT(10X,4H KB=,I3,10X,5H SUM=,F10.6/)
        IF(SUM-ET)480,480,410
C
C *** CALCULATE VAPOUR AND LIQUID ENTHALPIES FOR EACH PLATE
C
410     DO 430   J=1,N
416     TY3(J)=T(J)
418     DO 420 I=1,NOCOMP
420     TRANSFER(NL+I)=Y(I,J)
        TRANSFER(2)=T(J)
        TRANSFER(4)=0.
        TRANSFER(5)=1.
        CALL HSAT(TRANSFER)
```

```
      HV(J)=TRANSFER(4)
      DO 425   I=1,NOCOMP
425   TRANSFER(NL+I)=X(I,J)
      TRANSFER(5)=0.
      CALL HSAT(TRANSFER)
      HL(J)=TRANSFER(4)
430   CONTINUE
      TOTH=0.0
      DO 440 I=1,N
440   TOTH=TOTH+F(I)*HF(I)
      Q(1)=(HV(2)-HL(1))*V(2)+F(1)*(HF(1)-HL(1))-(HV(1)-HL(1))*(V(1)+W(1
     1))
      Q(N)=Q(1)+HL(N)*BB+HL(1)*U(1)+HV(1)*V(1)-TOTH
      RL(1)=V(2)-V(1)-U(1)
      SUM=0.
      DO 450 J=2,N-1
      V(J+1)=((HV(J)-HL(J))*(V(J)+W(J))+(HL(J)-HL(J-1))*RL(J-1)-F(J)*(HF
     1(J)-HL(J))+Q(J))/(HV(J+1)-HL(J))
      SUM=SUM+F(J)-W(J)-U(J)
450   RL(J)=SUM-DD+V(J+1)
      RL(N)=BB
C     DO 460 I=1,N
C460   WRITE(NW,1030)(I,T(I),RL(I),V(I),(X(J,I),J=1,NOCOMP))
      IF(KB.LT.15)GO TO 120
C
C
C
C *** OPTIONAL REBOILER MODEL.
C *** THIS MODEL IS USED ONLY IF A THIRD INPUT STREAM IS SPECIFIED.
C *** THE ROUTINE DEPLETES IT FROM THE HEAT LOAD NEEDED BY THE
C *** REBOILER AND OUTPUTS A COOLER STREAM AS THE THIRD OUTPUT
C *** STREAM.
C
C
480   IF(NIN-2)550,550,500
500   DO 510 I=1,NSLMAX
      STRMO(3,I)=STRMI(3,I)
510   TRANSFER(I)=STRMI(3,I)
      STRMO(3,3)=STRMO(3,3)-PARAM(9)
      TRANSFER(3)=STRMO(3,3)
```

```
7000  FORMAT(10X,'DISCOL/TRANSFER',8F12.3)
      CALL TEMP(TRANSFER)
      STRMO(3,2)=TRANSFER(2)
      IF(STRMO(3,2).GE.T(N))GO TO 550
      WRITE(NW,520)
520   FORMAT(/10X,34H ** WARNING FROM ROUTINE DISCOL ** /
     *10X,50H THE HEAT LOAD OF THE INPUT STREAM TO THE REBOILER /
     *10X,39H IS NOT SUFFICIENT TO HEAT THE REBOILER /
     *10X,43H THE SYSTEM OUTLET TEMPERATURE IS SET EQUAL /
     *10X,55H TO THE BOTTOMS TEMPERATURE BY DEFAULT, BUT THE ANSWERS /
     *10X,10H ARE WRONG /)
      STRMO(3,2)=T(N)
      TRANSFER(2)=T(N)
      TRANSFER(5)=-1.0
      CALL ENTHALPY(TRANSFER)
550   STRMO(3,4)=TRANSFER(4)
C
C *** DATA OUTPUT SECTION.
C
      STRMO(3,5)=TRANSFER(5)
      STRMO(1,1)=DD*(R+1.0)
      STRMO(1,2)=T(1)
      STRMO(1,3)=PRES
      STRMO(1,4)=(HV(1)*V(1)+HL(1)*U(1))/DD
      STRMO(1,5)=V(1)/DD
      DO 560 I=1,NOCOMP
      STRMO(1,NL+I)=(V(1)*Y(I,1)+U(1)*X(I,1))/DD
560   STRMO(2,NL+I)=X(I,N)
      STRMO(2,1)=BB
      STRMO(2,2)=T(N)
      STRMO(2,3)=PRES
      STRMO(2,4)=HL(N)
      STRMO(2,5)=0.
600   FORMAT(/10X,26H ** WARNING FROM DISCOL **    /
     110X,55H THE HEAT CONTENT OF THE HEATING STREAM TO THE REBOILER    /
     210X,40H IS NOT SUFFICIENT TO REBOIL THE BOTTOMS    /)
570   RETURN
1000  FORMAT(1H0,///,10X,15HL EXCEEDS LIMIT,//,5X,F10.5)
1010  FORMAT(1H1,8(/),10X,17H ITERATION NUMBER,I5,///)
1030  FORMAT(1H0,5X,I3,7F15.4/9X,7F15.4)
1040  FORMAT(1H0,5X,15H CONDENSER HEAT,F16.6,2X,6H BTU/H,//6X,14H REBOIL
```

```
     1ER HEAT,F17.6,2X,6H BTU/H/)
1050  FORMAT(1H1,8(/),42X,14HFINAL SOLUTION,///)
      END
C
C
      SUBROUTINE FLASHER
C
      DIMENSION TRANSFER(30)
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
     *,NW
C
C *** MODEL OF AN ADIABATIC FLASH TOWER.
C *** THE INPUT STREAM IS DIVIDED INTO ITS PURE PHASE COMPONENTS
C *** AT ITS PRESET OUTPUT PRESSURE.
C *** PURE PHASE STREAMS ARE OUTPUT IN THE APPROPRIATE STREAM.
C *** OUTPUT 1 = VAPOUR PHASE,  OUTPUT 2 = LIQUID PHASE.
C
      IF(MODE.GT.0)GO TO 40
      IF(PARAM(1).EQ.0.0)GO TO 50
      DO 10 I=1,NSLMAX
10    TRANSFER(I)=STRMI(1,I)
      TRANSFER(3)=PARAM(1)
      TRANSFER(5)=-1.0
C *** CALCULATE STREAM'S TEMPERATURE AT ITS NEW PRESSURE
      CALL TEMP(TRANSFER)
C *** SEPARATE THE TWO PHASES AT THIS NEW TEMPERATURE AND PRESSURE.
      CALL FLASH(TRANSFER,PARAM(1))
      DO 20 I=1,NSLMAX
      STRMO(1,I)=TRANSFER(I)
      STRMO(2,I)=TRANSFER(NSLMAX+I)
20    TRANSFER(NSLMAX+I)=0.0
      CALL ENTHALPY(TRANSFER)
      STRMO(1,4)=TRANSFER(4)
      DO 30 I=1,NSLMAX
      TRANSFER(I)=STRMO(2,I)
30    TRANSFER(NSLMAX+I)=0.0
      CALL ENTHALPY(TRANSFER)
      STRMO(2,4)=TRANSFER(4)
40    RETURN
50    WRITE(NW,60)
```

```
      PAUSE ER
      END
C
C
      SUBROUTINE HEATER
C
      DIMENSION TRANSFER(30)
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
     *,NW
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS
C
C *** MODEL OF A HEATER OR A COOLER.
C *** IT SETS THE OUTLET TEMPERATURE OF A STREAM TO A PRESET
C *** VALUE AND REDUCES ITS PRESSURE BY AN OPTIONAL PRESSURE
C *** DROP VALUE.  IT THEN COMPUTES ITS NEW ENTHALPY
C *** AND VAPOUR FRACTION AND OUTPUT A COMPLETELY DEFINED STREAM.
C
      IF(MODE.GT.0)GO TO 90
      IF(STRMI(1,1).EQ.0.0)RETURN
      IF(PARAM(1).EQ.0.0)WRITE(NW,130)
      DO 10 I=1,NSLMAX
      TRANSFER(I)=STRMI(1,I)
10    STRMO(1,I)=STRMI(1,I)
      STRMO(1,2)=PARAM(1)
20    STRMO(1,3)=STRMI(1,3)-PARAM(2)
      TRANSFER(2)=STRMO(1,2)
      TRANSFER(3)=STRMO(1,3)
      TRANSFER(5)=-1.
      CALL ENTHALPY(TRANSFER)
      STRMO(1,4)=TRANSFER(4)
      STRMO(1,5)=TRANSFER(5)
      RETURN
90    HDIFF=ABS(STRMO(1,4)-STRMI(1,4))*STRMI(1,1)
100   WRITE(NW,110)HDIFF
110   FORMAT(/10X,30H HEAT LOAD ON HEATER/COOLER = ,1E15.7,
     *' ENTHALPY UNITS')
120   RETURN
130   FORMAT(/10X,34H ** MESSAGE FROM ROUTINE HEATER ** /
     *10X,56H IS THE OUTLET TEMPERATURE SET TO ZERO, OR IS IT UNSET ?/)
      END
C
```

```
C
      SUBROUTINE HEATEXCH
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
     *.NW
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS
      DIMENSION TRANSFER(30)
      IF (MODE.GT.0)GO TO 445
C
      DO 15 J=1,2
      DO 10 I=1,NSLMAX
10    STRMO(J,I)=STRMI(J,I)
15    STRMO(J,3)=STRMI(J,3)-PARAM(J+3)
      IF(PARAM(3).GT.0.0)GO TO 300
C
C *** OUTLET TEMPERATURE OF FIRST STREAM GIVEN
C
      IF(PARAM(3).GT.0.0)GO TO 300
      IF(PARAM(1).EQ.0.0)WRITE(NW,500)
      TOUT=PARAM(1)
      IF(PARAM(6).GT.0.0)TOUT=TOUT+STRMI(2,2)
20    IF(STRMI(1,2)-STRMI(2.2))50,110,30
30    IF(TOUT-STRMI(1,2))40.110,160
40    IF(TOUT-STRMI(2,2))180,140,70
50    IF(TOUT-STRMI(1,2))200,110,60
60    IF(TOUT-STRMI(2,2))70.140.220
70    STRMO(1,2)=TOUT
75    STRMO(1,5)=-1.
      DO 80 I=1,NSLMAX
80    TRANSFER(I)=STRMO(1,I)
      CALL ENTHALPY(TRANSFER)
      STRMO(1,4)=TRANSFER(4)
      STRMO(1,5)=TRANSFER(5)
      HDIFF=(STRMI(1,4)-STRMO(1,4))*STRMI(1,1)
      STRMO(2,4)=STRMI(2,4)+HDIFF/STRMI(2,1)
      STRMO(2,5)=-1.
      DO 90 I=1,NSLMAX
90    TRANSFER(I)=STRMO(2,I)
      CALL TEMP(TRANSFER)
      STRMO(2,2)=TRANSFER(2)
```

```
94    IF(STRMO(2,2)-STRMI(1,2))100,100,96
96    STRMO(2,2)=STRMI(1,2)
      STRMO(2,5)=-1.
      DO 98 I=1,NSLMAX
98    TRANSFER(I)=STRMO(2,I)
      CALL ENTHALPY(TRANSFER)
      STRMO(2,4)=TRANSFER(4)
      STRMO(2,5)=TRANSFER(5)
      HDIFF=(STRMI(2,4)-STRMO(2,4))*STRMI(2,1)
      STRMO(1,4)=STRMI(1,4)+HDIFF/STRMI(1,1)
      STRMO(1,5)=-1.
      DO 100 I=1,NSLMAX
100   TRANSFER(I)=STRMO(1,I)
      CALL TEMP(TRANSFER)
      STRMO(1,2)=TRANSFER(2)
      STRMO(1,5)=TRANSFER(5)
108   HDIFF=ABS(HDIFF)
      GO TO 440
110   WRITE(NW,120)
120   FORMAT(/10X,51H T 1 OUT = T 1 IN , NO CHANGE IN STREAMS CONDITIONS
     1/)
      RETURN
140   WRITE(NW,150)
150   FORMAT(/10X,45H LARGE EXCHANGER REQUIRED AS T 1 OUT = T 2 IN /)
      GO TO 70
160   WRITE(NW,170)TOUT,STRMI(1,2)
170   FORMAT(/10X,39H IMPOSSIBLE SITUATION IN HEAT EXCHANGER,
     1/10X,10H T 1 OUT (,F10.2,32H DEG.) IS SET HIGHER THAN T 1 IN
     2/10X,2H (,F10.2,28H DEG.) FOR THE HOTTER STREAM
     3/10X,23H PLEASE CORRECT GUESSES/)
      PAUSE ER
180   WRITE(NW,190)TOUT,STRMI(2,2)
190   FORMAT(/10X,39H IMPOSSIBLE SITUATION IN HEAT EXCHANGER,
     1/10X,34H ATTEMPTING TO COOL HOT STREAM TO ,F10.2,5H DEG.
     2/10X,24H WHICH IS BELOW T 2 IN (,F10.2,6H DEG.)
     3/10X,23H PLEASE CORRECT GUESSES/)
      PAUSE ER
200   WRITE(NW,210)TOUT,STRMI(1,2)
210   FORMAT(/10X,39H IMPOSSIBLE SITUATION IN HEAT EXCHANGER,
     1/10X,10H T 1 OUT (,F10.2,32H DEG.) IS SET LOWER THAN T 1 IN
     2/10X,2H (,F10.2,28H DEG.) FOR THE COOLER STREAM
```

```
     3/10X,23H PLEASE CORRECT GUESSES/)
      PAUSE ER
220   WRITE(NW,230)TOUT,STRMI(2,2)
230   FORMAT(/10X,39H IMPOSSIBLE SITUATION IN HEAT EXCHANGER,
     1/10X,35H ATTEMPTING TO HEAT COLD STREAM TO ,F10.2,5H DEG.
     2/10X,24H WHICH IS ABOVE T 2 IN (,F10.2,6H DEG.)
     3/10X,23H PLEASE CORRECT GUESSES/)
      PAUSE ER
C
C *** TOTAL TUBE AREA AND TRANSFER COEFFICIENT (U) GIVEN
C
300   HDIFF=ABS(STRMI(1,2)-STRMI(2,2))*PARAM(2)*PARAM(3)/2.0
      FL=STRMI(1,1)
      IF(FL.GT.STRMI(2,1))FL=STRMI(2,1)
      IF((HDIFF/FL).GT.3000.)HDIFF=3000.*FL
305   HDIFF1=HDIFF
      IF(STRMI(1,2)-STRMI(2,2))320,110,310
310   HDIFF=-HDIFF
320   STRMO(1,4)=STRMI(1,4)+HDIFF/STRMI(1,1)
      STRMO(2,4)=STRMI(2,4)-HDIFF/STRMI(2,1)
      DO 340 J=1,2
      DO 330 I=1,NSLMAX
330   TRANSFER(I)=STRMO(J,I)
      TRANSFER(5)=-1.
      CALL TEMP(TRANSFER)
      STRMO(J,2)=TRANSFER(2)
      STRMO(J,5)=TRANSFER(5)
340   CONTINUE
      WRITE(2,1000)STRMO(1,2),STRMO(2,2),STRMO(1,4),
     *STRMO(2,4),HDIFF
1000   FORMAT(10X,'TS',2F15.4/10X,'HS',2F15.4/10X,'HDIFF',F15.4)
      DT1=STRMO(2,2)-STRMI(1,2)
      DT2=STRMI(2,2)-STRMO(1,2)
      IF(ABS(DT2-DT1)-0.005*ABS(DT1))350,350,360
350   DTLOG=ABS((DT1+DT2)/2.)
      GO TO 370
360   DTLOG=(DT2-DT1)/ALOG(DT2/DT1)
370   HDIFF=PARAM(2)*PARAM(3)*DTLOG
      WRITE(2,1010)DTLOG,HDIFF
1010  FORMAT(10X,'DTLOG,HDIFF',2F15.4/)
```

```
400   IF(STRMO(2,2)-STRMI(1,2))430,440,440
410   IF(STRMO(1,2)-STRMI(2,2))430,420,420
420   IF(STRMO(2,2)-STRMI(1,2))440,440,430
430   STRMO(1,2)=STRMI(2,2)
      GO TO 75
440   RETURN
445   HDIFF=ABS(STRMO(1,4)-STRMI(1,4))*STRMI(1,1)
450   WRITE(NW,460)HDIFF
460   FORMAT(/10X,31H HEAT LOAD IN HEAT EXCHANGER = ,1E15.7,
     +' ENTHALPY UNITS')
470   RETURN
500   FORMAT(/10X,36H ** MESSAGE FROM ROUTINE HEATEXCH ** /
     +10X,56H IS THE OUTLET TEMPERATURE SET TO ZERO, OR IS IT UNSET ?/)
      END
C
C
      SUBROUTINE MIXER
C
      DIMENSION TRANSFER(30)
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
     *,NW
C
C *** MODEL OF A MASS AND HEAT MIXER.
C *** INPUT STREAMS MAY HAVE DIFFERENT PRESSURES AND LOWEST ONE
C *** IS SET AS OUTPUT PRESSURE.
C
      IF(MODE.GT.0)GO TO 110
      DO 10 I=1,NSLMAX
10    STRMO(1,I)=0.
      DO 15 I=1,NIN
15    STRMO(1,1)=STRMO(1,1)+STRMI(I,1)
      IF(STRMO(1,1).EQ.0.0)RETURN
      IF(NOCOMP.EQ.0)GO TO 50
      DO 20 I=1,NIN
      STRMO(1,4)=STRMO(1,4)+STRMI(I,4)*STRMI(I,1)
      DO 20 J=1,NOCOMP
      STRMO(1,NL+J)=STRMO(1,NL+J)+STRMI(I,NL+J)*STRMI(I,1)
20    CONTINUE
      DO 30 J=1,NOCOMP
30    STRMO(1,NL+J)=STRMO(1,NL+J)/STRMO(1,1)
```

```
40    STRMO(1,4)=STRMO(1,4)/STRMO(1,1)
50    I1=0
      STRMO(1,3)=STRMI(1,3)
      DO 90 I=2,NIN
      IF(STRMO(1,3).EQ.STRMI(I,3))GO TO 90
      IF(I1.EQ.1)GO TO 80
      IF(ITRACE.NE.0)GO TO 80
60    WRITE(NW,70)
70    FORMAT(/10X,33H ** MESSAGE FROM ROUTINE MIXER **
     */10X,42H INPUTS TO MIXER HAVE DIFFERENT PRESSURES  /
     110X,32H LOWEST PRESSURE SET FOR OUTPUT /)
      I1=1
80    IF(STRMO(1,3)-STRMI(I,3))90,90,85
85    STRMO(1,3)=STRMI(I,3)
90    CONTINUE
      IF(NOCOMP.EQ.0)RETURN
      DO 100 I=1,NSLMAX
100   TRANSFER(I)=STRMO(1,I)
      TRANSFER(5)=-1.
      CALL TEMP(TRANSFER)
      STRMO(1,2)=TRANSFER(2)
      STRMO(1,5)=TRANSFER(5)
110   RETURN
      END
C
C
      SUBROUTINE PUMP
C
      DIMENSION TRANSFER(30)
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
     *,NW
C
C *** MODEL OF A PUMP OPERATING ISOTHERMALLY ON A
C *** LIQUID INPUT.  FOR GASES THE COMPRESR MODEL SHOULD BE USED.
C
      IF(MODE.GT.0)GO TO 20
      IF(PARAM(1).EQ.0.0)GO TO 50
      IF(STRMI(1,1).EQ.0.0)RETURN
      DO 15 I=1,NSLMAX
10    STRMO(1,I)=STRMI(1,I)
```

```
      TRANSFER(5)=-1.
      CALL ENTHALPY(TRANSFER)
      STRMO(1,4)=TRANSFER(4)
      STRMO(1,5)=TRANSFER(5)
      IF(STRMO(1,5).EQ.STRMI(1,5))RETURN
      STRMO(1,4)=STRMI(1,4)
      STRMO(1,5)=-1.
      DO 16 I=1,NSLMAX
16    TRANSFER(I)=STRMO(1,I)
      CALL TEMP(TRANSFER)
      STRMO(1,2)=TRANSFER(2)
      STRMO(1,5)=TRANSFER(5)
      RETURN
20    IF(PARAM(2).EQ.0.0)PARAM(2)=1.0
      HDIFF=ABS(STRMO(1,4)-STRMI(1,4))*STRMI(1,1)/PARAM(2)
      WRITE(NW,30)HDIFF
30    FORMAT(/10X,32H IDEAL ENERGY REQUIRED BY PUMP = ,1E15.7,
     *' ENTHALPY UNITS')
40    RETURN
50    WRITE(NW,60)
60    FORMAT(//10X,32H ** WARNING FROM ROUTINE PUMP ** /
     *10X,47H OUTLET PRESSURE NOT SET, SIMULATION ABANDONNED //)
      PAUSE ER
      END
C
C
      SUBROUTINE PURIFYER
C
      DIMENSION TRANSFER(30)
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
C
C *** MODEL OF A PURIFIER.
C *** COMPONENTS TO BE COMPLETELY REMOVED FROM A STREAM
C *** ARE SET IN THE SECOND OUTPUT STREAM FROM THIS UNIT  AND
C *** A PURIFIED STREAM IS OUTPUT AS THE FIRST OUTPUT STREAM.
C *** THE MODEL OUTPUTS THE TWO STREAMS COMPLETELY DEFINED.
C
      IF(MODE.GT.0)GO TO 80
      IF(STRMI(1,1).EQ.0.0)RETURN
      DO 10 I=1,NSLMAX
```

```
      STRMO(2,I)=0.
10    STRMO(1,I)=STRMI(1,I)
      M=PARAM(1)+0.1
      IF(M.LT.1)GO TO 80
      SUM=0.
      DO 20 I=1,M
      J=PARAM(I+1)
      STRMO(1,NL+J)=0.
      STRMO(2,NL+J)=STRMI(1,1)*STRMI(1,NL+J)
      SUM=SUM+STRMO(2,NL+J)
20    CONTINUE
      IF(SUM.EQ.0.0)GO TO 80
      DO 30 I=1,M
      J=PARAM(I+1)
30    STRMO(2,NL+J)=STRMO(2,NL+J)/SUM
35    STRMO(1,1)=STRMI(1,1)-SUM
      STRMO(2,1)=SUM
      STRMO(2,2)=STRMI(1,2)
      STRMO(2,3)=STRMI(1,3)
      STRMO(2,5)=STRMI(1,5)
      DO 40 I=1,NOCOMP
40    STRMO(1,NL+I)=STRMO(1,NL+I)*STRMI(1,1) /STRMO(1,1)
50    DO 60 I=1,NSLMAX
60    TRANSFER(I)=STRMO(1,I)
      CALL ENTHALPY(TRANSFER)
      STRMO(1,4)=TRANSFER(4)
      DO 70 I=1,NSLMAX
70    TRANSFER(I)=STRMO(2,I)
      CALL ENTHALPY(TRANSFER)
      STRMO(2,4)=TRANSFER(4)
80    RETURN
      END
C
C
      SUBROUTINE REACTOR
C
C     MODEL OF A STOICHIOMETRIC REACTOR
C
      DIMENSION TRANSFER(30)
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS
```

```
      IF(MODE.GT.0)GO TO 80
      IF(STRMI(1,1).EQ.0.0)RETURN
      FACTOR=0.1E10
      EXTENT=PARAM(NOCOMP+1)
      IFF=0
      IF(EXTENT.EQ.0.0)EXTENT=1.0
      IF(NIN.LT.2)GO TO 15
C *** MIX INPUTS IF THERE ARE MORE THAN ONE TO THE REACTOR
      CALL MIXER
      DO 10 I=1,NSLMAX
      TRANSFER(I)=STRMO(1,I)
10    STRMO(1,I)=0.0
      GO TO 18
15    DO 16 I=1,NSLMAX
16    TRANSFER(I)=STRMI(1,I)
18    DO 20 I=1,NOCOMP
      IF(PARAM(I).GE.0.0)GO TO 20
      IFF=1
      F=-TRANSFER(NL+I)/PARAM(I)
      IF(F.LT.FACTOR)FACTOR=F
20    CONTINUE
      IF(IFF.EQ.0)GO TO 70
      FACTOR=FACTOR*EXTENT*TRANSFER(1)
C *** MASS BALANCE BASED ON STOICHIOMETRIC COEFFICIENTS
      DO 30 I=1,NOCOMP
      TRANSFER(NL+I)=TRANSFER(NL+I)*TRANSFER(1)+FACTOR*PARAM(I)
30    SUM=SUM+TRANSFER(NL+I)
      DO 40 I=1,NOCOMP
40    TRANSFER(NL+I)=TRANSFER(NL+I)/SUM
      FLRATE=TRANSFER(1)
      TRANSFER(1)=SUM
      TRANSFER(4)=(TRANSFER(4)*FLRATE+PARAM(NOCOMP+2)*FACTOR)/SUM
      TRANSFER(5)=-1.0
C *** CALCULATE THE NEW OUTPUT TEMPERATURE.
      CALL TEMP(TRANSFER)
      DO 50 I=1,NSLMAX
50    STRMO(1,I)=TRANSFER(I)
60    RETURN
70    WRITE(NW,80)
80    FORMAT(/10X,'*** ERROR IN ROUTINE REACTOR ***'/
     *10X,'ALL STOICHIOMETRIC FACTORS ARE POSITIVE IN PARAMETER ARRAY'/
```

```
     *10X,'SITUATION IMPOSSIBLE,  SIMULATION ABANDONNED ***'/)
      PAUSE ER
      END
C
C
      SUBROUTINE SETBP
C
      DIMENSION TRANSFER(30)
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
C
C *** THIS ROUTINE SETS THE TEMPERATURE OF A STREAM
C *** TO ITS BUBBLE POINT BASED ON ITS INPUT PRESSURE.
C
      IF(MODE.GT.0)RETURN
      IF(STRMI(1,1).EQ.0.0)RETURN
      DO 10 I=1,NSLMAX
      TRANSFER(I)=STRMI(1,I)
10    STRMO(1,I)=STRMI(1,I)
      TRANSFER(5)=0.0
      IRL=1
C *** CALCULATE BUBBLE POINT.
      CALL DEWBUB(IRL,TRANSFER,STRMO(1,2))
      STRMO(1,5)=0.0
      TRANSFER(2)=STRMO(1,2)
C *** CALCULATE NEW ENTHALPY
      CALL ENTHALPY(TRANSFER)
      STRMO(1,4)=TRANSFER(4)
      RETURN
      END
C
C
      SUBROUTINE SETDP
C
      DIMENSION TRANSFER(30)
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
C
C *** THIS ROUTINE SETS THE TEMPERATURE OF A STREAM TO ITS
C *** DEW POINT BASED ON ITS INPUT PRESSURE.
C
      IF(MODE.GT.0)RETURN
```

```
10    STRMO(1,1)=STRMI(1,1)
      TRANSFER(5)=1.
      IRL=0
C *** CALCULATE DEW POINT.
      CALL DEWBUB(IRL,TRANSFER,STRMO(1,2))
      STRMO(1,5)=1.
      TRANSFER(2)=STRMO(1,2)
C *** CALCULATE NEW ENTHALPY.
      CALL ENTHALPY(TRANSFER)
      STRMO(1,4)=TRANSFER(4)
      RETURN
      END
C
C
      SUBROUTINE SPLITTER
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
     *,NW
C
C *** MODEL OF A STREAM DIVIDER (WITH UP TO 6 OUTPUTS).
C *** THE SPLIT FACTORS ARE SET IN THE PARAMETER ARRAY AND
C *** NORMALISED IN THE ROUTINE.
C
      IF(MODE.GT.0)GO TO 50
      IF(STRMI(1,1).EQ.0.0)RETURN
      SUM=0.
      DO 10 I=1,NOUT
10    SUM=SUM+PARAM(I)
      IF(SUM.EQ.0.0)GO TO 60
      DO 20 I=1,NOUT
20    PARAM(I)=PARAM(I)/SUM
      DO 30 I=1,NOUT
      DO 30 J=2,NSLMAX
30    STRMO(I,J)=STRMI(1,J)
      DO 40 I=1,NOUT
40    STRMO(I,1)=STRMI(1,1)*PARAM(I)
50    RETURN
60    WRITE(NW,70)
70    FORMAT(//10X,36H ** WARNING FROM ROUTINE SPLITTER ** /
     *10X,53H NO SPLITTING FACTORS HAVE BEEN SET IN THE PARAMETERS /
     *10X,30H THE SIMULATION IS ABANDONNED  //)
```

```
      PAUSE ER
      END
C
C
      SUBROUTINE SPLITMIX
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
C
C *** MODEL OF A MULTI-INPUT, MULTI-OUTPUT HOLDING TANK.
C *** THE OUTPUTS ARE SPLIT ACCORDING TO THE SPLIT FACTORS SET
C *** IN THE PARAMETER ARRAY.
C
      IF(MODE.GT.0)RETURN
      CALL MIXER
      DO 10 I=1,NSLMAX
10    STRMI(1,I)=STRMO(1,I)
      CALL SPLITTER
      RETURN
      END
C
C
      SUBROUTINE VALVE
C
      DIMENSION TRANSFER(30)
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX,NL
     *,NW
C
C *** MODEL OF AN ADIABATIC THROTTLE VALVE.
C *** THE PRESSURE OF THE INPUT STREAM IS REDUCED TO THE DESIRED
C *** OUTPUT PRESSURE AND THE OUTPUT STREAM'S NEW TEMPERATURE
C *** AND VAPOUR FRACTION ARE CALCULATED BASED ON ITS CONSTANT
C *** ENTHALPY AND COMPOSITION.
C
      IF(MODE.GT.0)GO TO 20
      IF(STRMI(1,1).EQ.0.0)RETURN
      IF(PARAM(1).EQ.0.0)GO TO 30
      DO 10 I=1,NSLMAX
      STRMO(1,I)=STRMI(1,I)
10    TRANSFER(I)=STRMI(1,I)
      TRANSFER(3)=PARAM(1)
```

```
      CALL TEMP(TRANSFER)
      STRMO(1,2)=TRANSFER(2)
      STRMO(1,5)=TRANSFER(5)
20    RETURN
30    WRITE(NW,40)
40    FORMAT(//10X,33H ** WARNING FROM ROUTINE VALVE ** /
     *10X,46H NO VALVE PRESSURE HAS BEEN SET IN PARAMETER 1 /
     *10X,30H THE SIMULATION IS ABANDONNED //)
      PAUSE ER
      END
C
C
      SUBROUTINE MASSMIX
C
      COMMON MODE,NIN,NOUT,STRMI(6,15),STRMO(6,15),PARAM(30),NSLMAX
C
C *** MODEL OF A MASS ONLY MIXER.
C *** THIS MODEL IS USED FOR MASS BALANCES ONLY.
C *** IT SETS THE TEMPERATURE AND ENTHALPY TO THEIR AVERAGE
C *** VALUE AND THE PRESSURE TO ITS LOWEST INPUT VALUE.
C
      IF(MODE.GT.0)RETURN
      DO 10 I=1,NSLMAX
10    STRMO(1,I)=0.0
      STRMO(1,3)=STRMI(1,3)
      DO 30 I=1,NIN
      STRMO(1,1)=STRMO(1,1)+STRMI(I,1)
      IF(STRMO(1,3).GT.STRMI(I,3))STRMO(1,3)=STRMI(I,3)
      DO 20 J=2,NSLMAX
      IF(J.EQ.3)GO TO 20
      STRMO(1,J)=STRMO(1,J)+STRMI(I,J)*STRMI(I,1)
20    CONTINUE
30    CONTINUE
C *** AVERAGING TEMPERATURE, ENTHALPY AND COMPOSITION.
      IF(STRMO(1,1).LE.0.0)RETURN
      DO 40 J=2,NSLMAX
      IF(J.EQ.3)GO TO 40
      STRMO(1,J)=STRMO(1,J)/STRMO(1,1)
40    CONTINUE
      RETURN
      END
```

Program PHYSUBS

```
      LIST(LP)
      SEGMENTS(FXXX)
      INPUT 1 = CR0
      OUTPUT 2 = LP0
      COMPRESS INTEGER AND LOGICAL
      COMPACT
      TRACE 2,500
      END
      FUNCTION DENSTY(TRANSFER)
C
C     DENSITY IS GIVEN IN KG PER M**3 OR LB PER FT**3
C
      DIMENSION
     1 Z(30),TRANSFER(30),X(10),CFACT(5,7),PRC(7)
C
      COMMON
     1/PROPS/PROPRT(10,25),US(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL,NW
     2/PROPE/PRTEXTRA(10,32)
     3/RK    /BIGA(10),BIGB(10),AABAR,BBBAR,ABAR
C
      DATA CFACT /
     *0.1220590E01, 0.5947060E00,-0.3402229E01, 0.4276129E01,-0.2141768
     *E01,
     *0.1152644E01, 0.1189276E01,-0.5170992E01, 0.6464711E01,-0.3066619
     *E01,
     *0.1317513E01,-0.1678128E00,-0.1154578E01, 0.1466986E01,-0.7659117
     *E00,
     *0.1305978E01,-0.7567871E-1,-0.1352292E01, 0.1604660E01,-0.7173434
     *E00,
     *0.1404976E01,-0.8075938E00, 0.5723827E00,-0.4530966E00, 0.9028040
     *E-1,
     *0.1359969E01,-0.4536306E00,-0.3588908E00, 0.5897938E00,-0.3145051
     *E00,
     *0.1364005E01,-0.4097534E00,-0.5246663E00, 0.7552146E00,-0.3151541
     *E00
     */
C
      DATA PRC/
     *0.6,1.0,3.0,5.0,8.0,10.0,15.0
```

```
     +TR*CFACT(3/2))))
C
C      CONVERSION FACTORS
C
       IF(IUNITS.GT.0)GO TO 10
       R=10.7335
       TADD=459.7
       PRES1=14.7
       FACT1=62.4
       FACT2=1.0
       FACT3=1.8
       GO TO 20
10     R=0.084778
       TADD=273.15
       PRES1=1.0
       FACT1=1.0E03
       FACT2=1.8
       FACT3=1.0
C
20     DENSTY=0.0
       K=0
       DO 30 I=1,NSLMAX
30     Z(I)=TRANSFER(I)
       IF(Z(5).EQ.0.0)GO TO 80
       IF(Z(5).EQ.1.0)GO TO 40
       CALL FLASH(Z,Z(3))
       TRANSFER(5)=Z(1)/TRANSFER(1)
       IF(TRANSFER(5).LT.0.005)GO TO 80
C
C      DENSITY OF THE VAPOUR PHASE
C      EVALUATION OF THE COMPRESSIBILITY FACTOR BY THE REDLICH-KWONG
C      EQUATION OF STATE
C
40     W=0.0
       DO 50 I=1,NOCOMP
       W=W+Z(NL+I)*PROPRT(I,4)
50     X(I)=Z(NL+I)
       T=Z(2)
       CALL RKDATA(X,T)
       C=AABAR*AABAR/BBBAR
       X2=0.9
```

```
60    X1=X2
      X12=X1*X1
      X13=X1*X12
      BZ=BBBAR*Z(3)
      C1=X13-X12+BZ*(C-BZ-1.0)*X1-C*BZ*BZ
      C2=3.0*X12-2.0*X1+BZ*(C-BZ-1.0)
      X2=X1-C1/C2
      IF(ABS(X2-X1).GT.0.001)GO TO 60
      DENSV=Z(3)*W/(R*X2*(Z(2)+TADD))
      IF(TRANSFER(5).LT.0.995)GO TO 75
C *** DENSITY HERE IS IN THE UNITS OF THE GAS CONSTANT
C *** LB/FT**3  OR  GM/LT (=KG/M**3)
      DENSTY=DENSV
      RETURN
C
C     DENSITY OF THE LIQUID PHASE
C
75    K=NSLMAX
80    RNOM=0.0
      DENOM=0.0
      T=(Z(K+2)+TADD)*FACT2
      DO 210 I=1,NOCOMP
      IF(Z(K+NL+I).LT.0.001)GO TO 210
      IF(PRTEXTRA(I,1))115,170,90
90    KALL=1
C *** PRTEXTRA(I,1) SHOULD BE IN DEGREES R.
      IF(T.LE.PRTEXTRA(I,1))GO TO 110
      DENSL=(PRTEXTRA(I,6)*(PROPRT(I,6)*FACT2-T))**
     *(1.0/PRTEXTRA(I,7))+PROPRT(I,4)/PROPRT(I,7)/FACT1
      GO TO 120
110   T=T-459.7
      DENSL=PRTEXTRA(I,2)-PRTEXTRA(I,5)*T-PRTEXTRA(I,4)/
     /(PRTEXTRA(I,5)-T)
      T=T+459.7
      GO TO 120
C
115   T2=-PRTEXTRA(I,1)
      IF(T2.GT.T)GO TO 110
      T=T2
      GO TO 110
```

```
130    DO 135 L=2,7
       IF(PR.LT.PRC(L))GO TO 140
135    CONTINUE
140    L=L-1
       C1=F(L,TR)
       IF(L.EQ.1)GO TO 150
       L=L+1
145    C2=F(L,TR)
       C1=(C1-C2)/(PRC(L)-PRC(L-1))*(PR-PRC(L-1))+C1
150    GO TO(155,160),KALL
155    PR=Z(3)/PROPRT(I,5)
       TR=(Z(2)+TADD)/PROPRT(I,6)
       CC1=C1
       KALL=2
       GO TO 130
160    DENSL=DENSL*C1/CC1
       GO TO 200
C
C ***  PRTEXTRA(I,3) SHOULD BE IN DEGREES F.
C ***  PRTEXTRA(I,2) SHOULD BE IN GM/CC
170    T=(PRTEXTRA(I,3)+TADD)
       DENSL=PRTEXTRA(I,2)
       TR=T/PROPRT(I,6)
       GO TO 125
C
200    RNOM=RNOM+Z(K+NL+I)*PROPRT(I,4)
       DENOM=DENOM+Z(K+NL+I)*PROPRT(I,4)/DENSL
210    CONTINUE
       DENSL=RNOM/DENOM*FACT1
       IF(TRANSFER(5).NE.0.0)GO TO 220
       DENSTY=DENSL
       RETURN
C
C      DENSITY OF TWO PHASE MIXTURE
C
220    X2=0.0
       X1=0.0
       DO 230 I=1,NOCOMP
       X2=X2+Z(NL+I)*PROPRT(I,4)
230    X1=X1+Z(K+NL+I)*PROPRT(I,4)
       X2=TRANSFER(5)*X2
```

```
      X1=(1.0-TRANSFER(5))*X1
      RNOM=X2+X1
      DENOM=X2/DENSV+X1/DENSL
      DENSTY=RNOM/DENOM
      RETURN
      END
      FUNCTION SPHEAT(TRANSFER)
C
C     SPECIFIC HEAT IS GIVEN IN KJOULE PER (KG,DEGREE C)
C     ARRAY TRANSFER SHOULD BE SUPPLIED IN BRITISH UNITS
C
      DIMENSION
     1 TRANSFER(30),Z(30),R(6,4)
      COMMON
     1/PROPS/PROPRT(10,25) ,US(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL,NW
     2/PROPE/PRTEXTRA(10,32)
C
C      CONVERSION FACTORS
C
      IF(IUNITS.GT.0)GO TO 3
      TADD=459.7
      RGAS=10.7335
      FACT1=1.0
      FACT2=1.0
      GO TO 5
3     TADD=273.15
      RGAS=0.084778
      FACT1=4.184
      FACT2=1.8
5     K=0
      DO 10 I=1,NSLMAX
   10 Z(I) = TRANSFER(I)
      VF=TRANSFER(5)
      IF(TRANSFER(5).EQ.0.) GO TO 75
      IF(TRANSFER(5).EQ.1.) GO TO 20
      CALL FLASH(Z,Z(3))
      TRANSFER(5)=Z(1)/TRANSFER(1)
      IF(TRANSFER(5).EQ.0.) GO TO 70
C
```

```
      B=0.
      C=0.
      W = 0.
      OMEGA = 0.
C     CALCULATION OF MIXTURE CORRESPONDENCE PROPERTIES
      DO 30 I=1,NOCOMP
      IF(Z(NL+I).LT. .001) GO TO 30
      A=A+Z(NL +I)*PROPRT(I,6)/SQRT(PROPRT(I,5))
      D=PROPRT(I,6)/PROPRT(I,5)
      B=B+Z(NL+ I)*D
      C=C+Z(NL+ I)*SQRT(D)
      W = W+Z(NL+I)*PROPRT(I,4)
      OMEGA=OMEGA+Z(NL+I)*PROPRT(I,9)
   30 CONTINUE
      B=B/3.+2.*C*C/3.
      TMC= A*A/B
      PMC= TMC/B
      ZCM= .291-.08*OMEGA
      DENSCM=PMC/(RGAS*ZCM*TMC)*W
C
C     REDUCED PROPERTIES
      TR=(Z(2)+TADD)/TMC
      PR = Z(3)/PMC
      DENSR = DENSTY(Z) /DENSCM
C
C     HEAT CAPACITY AT LOW PRESSURE
      A=(Z(2)+TADD)*FACT2/100.0
      CPV=0.0
      W  =0.
      DO 50 I=1,NOCOMP
      IF(Z(NL+I).LT.0.001)GO TO 50
      IF(PRTEXTRA(I,8).EQ.0.0)GO TO 40
      T=1.0
      CP=PRTEXTRA(I,11)/A
      DO 35 J=1,3
      T=T*A
35    CP=CP+PRTEXTRA(I,7+J)*T
      GO TO 48
40    TA=1.0
      A=Z(2)
      IF(IUNITS.GT.0)A=A*1.8+32.
```

```
      CP=PROPRT(I,15)
      DO 45 J=1,4
      TA=TA*A
45    CP=CP+TA*PROPRT(I,15+J)*FLOAT(J+1)
48    D = Z(NL+I)*PROPRT(I,4)
      W=W+D
      CPV=CPV+D*CP
   50 CONTINUE
      CPV=CPV/W
C
C     CORRECTION FOR PRESSURE
      B= (((2274.28027*ZCM-822.13757)*ZCM-153.6915)*ZCM+36.78284)*ZCM+
     +12.51368
      A= B-5.5
      C= 5.811+4.919*OMEGA
      D = .5*(2.*B-4.5-C)
      IF(DENSR.GE.1.) GO TO 55
C
C     REDUCED DENSITY LESS THAN 1.
      DCV= ZCM*(2.*A-D*DENSR)*DENSR/(TR*TR)
      B1 = (B-3.)/(3.*B-1.)
      B2 = ((3.*B-6.)*B-1.)/(3.*B*B-B)
      DPD= (-2.*(5.5+A/TR)+3.*D*(-TR+1./TR)*DENSR)*DENSR+TR*(1.-B1*DENSR
     R*DENSR)/(ZCM*(1.+(-B2+B1*DENSR)*DENSR)**2)
      DPT= ((-D*(1.+1./(TR*TR))*DENSR+A/(TR*TR))*DENSR+1./(ZCM*(1.+(-B2+
     +B1*DENSR)*DENSR)))*DENSR
      GO TO 65
C
C     REDUCED DENSITY GREATER THAN 1.
   55 R(1,1) =0.
      R(2,1) =0.
      R(3,1) =0.
      R(4,1) =5.5-B
      R(5,1) = -2.25-C/2. +B
      R(6,1) =0.
      R(1,2) =88.5-3.12*B
      R(2,2) =-313.3+13.42*B
      R(3,2) =408.9-21.54*B
      R(4,2) =-237.4+15.3*B
      R(5,2) =47.8-4.06*B
```

```
      R(3,3) =-320.8-(1.78+3.63*B)*B
      R(4,3) =263.+(15.7-3.63*B)*B
      R(5,3) =C/2.-27.05+(-16.18+1.815*B)*B
      R(6,3) =-8.44+(4.50-.365*B)*B
      R(1,4) =44.4-5.22*B
      R(2,4) =-156.9+18.92*B
      R(3,4) =204.3-25.44*B
      R(4,4) =-115.5+15.*B
      R(5,4) =23.7-3.26*B
      R(6,4) =0.
C
      B1 =0.
      DPD=0.
      DPT=0.
      DCV=-ZCM*(2.*A-D)/(TR*TR)
      DO 60 J=1,4
      A= ((.3333*R(6,J)*DENSR+.5*R(5,J))*DENSR+R(4,J))*DENSR+R(3,J)*ALOG
     G(DENSR)-(.5*R(1,J)/DENSR+R(2,J))/DENSR
      A=A-(-.5*R(1,J)-R(2,J)+R(4,J)+.5*R(5,J)+.333*R(6,J))
      B=((4.*R(6,J)*DENSR+3.*R(5,J))*DENSR+2.*R(4,J))*DENSR+R(3,J)-
     1R(1,J)/(DENSR*DENSR)
      C=(((R(6,J)*DENSR+R(5,J))*DENSR+R(4,J))*DENSR+R(3,J))*DENSR+
     +R(2,J)+R(1,J)/DENSR
      B1= B1+FLOAT((J-2)*(J-3))*A*TR**(J-3)
      DPD=DPD+B*TR**(J-2)
      DPT=DPT+FLOAT(J-2)*C*TR**(J-3)
   60 CONTINUE
      DCV=DCV+ZCM*B1
C
65    DCP=DCV-(ZCM*TR)*DPT*DPT/(DPD*DENSR*DENSR)+1.0
      CPV=CPV-1.9872*DCP/W
      IF(TRANSFER(5).NE.1.) GO TO 70
      SPHEAT=CPV
      GO TO 110
C
C     HEAT CAPACITY OF LIQUID PHASE
C
   70 K= NSLMAX
   75 W= 0.
      CPL=0.
      T=Z(K+2)
```

```
      IF(IUNITS.EQ.0)T=(T-32.)/1.8
      DO 85 I=1,NOCOMP
      IF(Z(NL+I).LT.0.001) GO TO 85
      CP =PRTEXTRA(I,12)
      TA=1.0
      DO 80 J=1,5
      TA=TA*T
80    CP=CP+T*PRTEXTRA(I,12+J)
      D =Z(K+NL+I)*PROPRT(I,4)
      CPL=CPL+CP*D
      W  =W+D
   85 CONTINUE
      CPL=CPL/W
      IF(TRANSFER(5).NE.0.) GO TO 90
      SPHEAT=CPL
      GO TO 110
C
C    SPECIFIC HEAT OF TWO-PHASE MIXTURE
C
90    WM=0.0
      WV=0.0
      WL=0.0
      DO 100 I=1,NOCOMP
      WM=WM+PROPRT(I,4)*TRANSFER(NL+I)
      WV=WV+PROPRT(I,4)*Z(NL+I)
100   WL=WL+PROPRT(I,4)*Z(NSLMAX+NL+I)
      SPHEAT=(CPV*WV*VF+CPL*WL*(1.0-VF))/WM
C
110   SPHEAT=FACT1*SPHEAT
      RETURN
      END
      FUNCTION VISC(TRANSFER )
C
C NOTE: VISCOSITY OF TWO-PHASE MIXTURE IS NOT EVALUATED
C     THE LIQUID VISCOSITY IS RETURNED INSTEAD
C     VISCOSITY IS GIVEN IN KG PER (METRE,SEC) OR  LB PER (FT,SEC)
C
      DIMENSION
     1 TRANSFER(30),Z(30),VSCF(6,3)
      COMMON
```

```
C
      DATA VSCF/-10.397841,79.28618,-180.737,173.8126,-61.45769,0.0,
     130.48008,-236.4402,695.1043,-1021.036,741.7547,-212.9425,
     2-1.188684,0.2047325,-9.88394,15.34354,-7.55261,0.0
     */
C
      VSC(TR,N)=VSCF(1,N)+TR*(VSCF(2,N)+TR*(VSCF(3,N)+TR*(VSCF(4,N)+
     +TR*(VSCF(5,N)+TR*VSCF(6,N)))))
C
C     CONVERSION FACTORS
C
      IF(IUNITS.GT.0)GO TO 5
      R=10.7335
      FACT1=1.
      FACT2=1.
      FACT3=6.72E-04
      FACT4=14.7
      FACT5=62.4
      FACT6=1.8
      TADD=459.7
      GO TO 10
5     R=0.084778
      FACT1=1.8
      FACT2=16.03
      FACT3=1.0E-03
      FACT4=1.0138
      FACT5=1.0E3
      FACT6=1.0
      TADD=273.15
C
10    K=0
      VISC=0.0
      DO 15 I=1,NSLMAX
15    Z(I)=TRANSFER(I)
      IF(Z(5).EQ.0.0)GO TO 110
      IF(Z(5).EQ.1.0)GO TO 20
      CALL FLASH(Z,Z(3))
      TRANSFER(5)=Z(1)/TRANSFER(1)
      IF(TRANSFER(5).GT.0.95)GO TO 20
      DO 16 I=1,NSLMAX
      Z(I)=Z(NSLMAX+I)
```

```
16    Z(NSLMAX+I)=0.0
C
C     VISCOSITY OF THE VAPOUR PHASE
C
20    W=0.0
      VISCV=0.0
      TABS=Z(2)+TADD
      IF(IUNITS.GT.0)TABS=TABS*1.8
      DO 70 I=1,NOCOMP
      IF(Z(NL+I).LT.0.001)GO TO 70
      IF(PRTEXTRA(I,18).GT.0.0)GO TO 25
      TC=PROPRT(I,6)*FACT1
      ZC=PROPRT(I,5)*PROPRT(I,7)/R/PROPRT(I,6)
      PRTEXTRA(I,18)=65.3*TC*ZC**3.6
      VC=PROPRT(I,7)*FACT2
      PRTEXTRA(I,19)=0.7402*VC**0.33333/ZC**1.2
25    A=TABS/PRTEXTRA(I,18)
      IF(A.LE.2.7) GO TO 45
      IF(A.LE.5.)  GO TO 40
      IF(A.LE.10.) GO TO 35
      IF(A.LE.100.)GO TO 30
C
      A=.01*A
      OMEGA = ((-.002658*A+.028859)*A-.124181)*A+.686182
      GO TO 50
   30 A=.1*A
      OMEGA =(((.000099*A-.002720)*A+.028061)*A-.142688)*A+.9401
      GO TO 50
   35 OMEGA = ((-.00026*A+.00792)*A-.09370)*A+1.22996
      GO TO 50
   40 OMEGA =(((.0004*A-.00997)*A+.09468)*A-.43390)*A+1.72599
      GO TO 50
   45 OMEGA =(((.12792*A-1.03864)*A+5.18405)*A-4.59165)*A+3.89543
C
C
C     THE MIXING RULE USED HERE APPLIES ONLY AT LOW PRESSURES (PR.LT.0.6)
C
50    XM=0.001989*SQRT(PROPRT(I,4)*TABS)/PRTEXTRA(I,19)**2/OMEGA
      X=Z(NL+I)*SQRT(PROPRT(I,4))
      VISCV=VISCV+XM*X
```

```
C
C     VISCOSITY OF THE LIQUID PHASE
C
110   VISCL=0.0
      TZ=Z(2)
      DO 120 I=1,NOCOMP
      TRANSFER(NSLMAX+NL+I)=Z(NL+I)
120   Z(NL+I)=0.0
      IF(IUNITS.EQ.0)TZ=(TZ-32.0)/1.8
      DO 200 I=1,NOCOMP
      IF(TRANSFER(NSLMAX+NL+I).LT.0.001)GO TO 200
      Z(NL+I)=1.0
      IF(PRTEXTRA(I,20).EQ.0.0)GO TO 125
      VIS=PRTEXTRA(I,20)
      T=1.0
      DO 122 J=1,6
      T=T*TZ
      IF(PRTEXTRA(I,20+J).EQ.0.0)GO TO 122
      VIS=VIS+T*PRTEXTRA(I,20+J)
122   CONTINUE
      GO TO 190
125   IF(Z(2).GT.(PROPRT(I,8)-TADD))GO TO 150
      Z(5)=0.0
      D=DENSTY(Z)/FACT5
      IF(PRTEXTRA(I,21).GT.1.0)GO TO 140
C
C     SOUDER'S METHOD FOR ESTIMATING LIQUID VISCOSITIES
C     AT TEMPERATURES BELOW THE NORMAL BOILING POINT
C
      VIS=D*PRTEXTRA(I,22)/PROPRT(I,4)-2.9
      VIS=EXP10(VIS)
      VIS=EXP10(VIS)/10.
      GO TO 190
C
C     THOMAS'S METHOD FOR ESTIMATING LIQUID VISCOSITIES
C     AT TEMPERATURES BELOW THE NORMAL BOILING POINT
C
140   TR1=PROPRT(I,6)/(Z(2)+TADD)-1.0
      VIS=PRTEXTRA(I,22)*TR1
      VIS=EXP10(VIS)*SQRT(D)/8.569
```

```
      GO TO 190
C
C     STIEL AND THODOS'S METHOD FOR ESTIMATING LIQUID VISCOSITIES
C     AT TEMPERATURES ABOVE THE NORMAL BOILING POINT
C
150   TR=(Z(2)+TADD)/PROPRT(I,6)
      TC=PROPRT(I,6)/FACT6
      PC=PROPRT(I,5)/FACT4
      E=PC**4.0
      E=(TC/E)**0.166666/SQRT(PROPRT(I,4))
      ZC=PROPRT(I,5)*PROPRT(I,7)/R/PROPRT(I,6)
      N=2
      V1=VSC(TR,N)
      N=1
      DZ=-0.02
      IF(ZC.LT.0.27)GO TO 160
      N=3
      DZ=0.02
160   V2=VSC(TR,N)
      VIS=(V2-V1)/DZ*(ZC-0.27)+V1
      VIS=VIS/E
190   VISCL=VISCL+TRANSFER(NSLMAX+NL+I)*VIS**0.333333
      Z(NL+I)=0.0
200   CONTINUE
      VISC=VISCL**3.0*FACT3
      RETURN
      END
      FUNCTION CNDVTY(TRANSFER)
C
C     THE THERMAL CONDUCTIVITY OF A TWO-PHASE MIXTURE IS NOT CALCULATED.
C     THE LIQUID PHASE CONDUCTIVITY ONLY IS RETURNED INSTEAD
C
      COMMON
     1/PROPS/ PROPRT(10,25),U(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL,NW
     2/PROPE/ PRTEXTRA(10,32)
C
      DIMENSION TRANSFER(30),Z(30)
C
      F1(X)=14.0*(EXP(-0.535*X)-1.0)*1.0E-08
```

```
C      CONVERSION FACTORS
C
       IF(IUNITS.GT.0)GO TO 10
       TADD=459.7
       TZERO=491.7
       R=10.7335
       PRES1=500.
C      14.7**4/1.8  =  .259416E05
       FACT1=0.259416E05
       FACT2=1.0
       FACT4=15.12
       FACT5=62.4
       GO TO 20
10     TADD=273.15
       TZERO=273.15
       R=0.084778
       PRES1=35.
       FACT1=1.05427
       FACT2=0.239
       FACT4=2.39
       FACT5=1000.
C
20     K=0
       CNDVTY=0.0
       RNOM=0.0
       DENOM=0.0
       DO 30 I=1,NSLMAX
30     Z(I)=TRANSFER(I)
       IF(Z(5).EQ.1.0)GO TO 40
       IF(Z(5).EQ.0.0)GO TO 120
       CALL FLASH(Z,Z(3))
       TRANSFER(5)=Z(1)/TRANSFER(1)
       IF(TRANSFER(5).GT.0.95)GO TO 40
C ***  IF THE MIXTURE CONTAINS LIQUID, TRANSCRIBE THE LIQUID SECTION
C ***  TO ARRAY Z(I)
       DO 35 I=1,NSLMAX
       Z(I)=Z(NSLMAX+I)
35     Z(NSLMAX+I)=0.0
       GO TO 120
C
C      THERMAL CONDUCTIVITY OF THE VAPOUR PHASE
```

```
C      MISIC AND THODOS'S METHOD
C
40     DO 50 I=1,NOCOMP
       TRANSFER(NSLMAX+NL+I)=Z(NL+I)
50     Z(NL+I)=0.0
       DO 100 I=1,NOCOMP
       IF(TRANSFER(NSLMAX+NL+I).LT.0.001)GO TO 100
       Z(NL+I)=1.0
       CP=SPHEAT(Z) *FACT2*PROPRT(I,4)
       TR=(Z(2)+TADD)/PROPRT(I,6)
       GAMMA=(PROPRT(I,6)/PROPRT(I,5)**4*FACT1)**0.166666*SQRT(PROPRT(I,
      ,4))
       ZC=PROPRT(I,5)*PROPRT(I,7)/R/PROPRT(I,6)
       K=PRTEXTRA(I,27)+0.1
       GO TO (55,60,65),K
55     IF(TR.GT.1.0)GO TO 60
       CNDV=0.445E-05*TR*CP/GAMMA
       GO TO 80
60     CNDV=1.0E-06*(14.52*TR-5.14)**0.666666*CP/GAMMA
       GO TO 80
65     IF(TR.LT.1.0)GO TO 75
       IF(TR.LT.3.0)GO TO 70
       CNDV=1.0E-05*((7.18-18.25*ZC)*TR+10.21*ZC-4.91)**(1.079-1.97*
      *ZC)*CP**0.75/GAMMA
       GO TO 80
70     CNDV=1.0E-06*((195.*ZC-31.94)*TR+16.83-82.5*ZC)**(1.524-2.8*ZC)*
      *CP**0.75/GAMMA
       GO TO 80
75     CNDV=1.0E-06*(20.0*ZC+1.08)*TR**(1.81-2.604*ZC)
80     IF(Z(3).LT.PRES1)GO TO 90
C ***  PRESSURE EFFECT CORRECTION ON THE THERMAL CONDUCTIVITY
       DR=DENSTY(Z)/PROPRT(I,4)*PROPRT(I,7)
       C=GAMMA*ZC**5
       IF(DR.LE.0.5)F=F1(DR)
       IF(DR.LT.2.0.AND.DR.GT.0.5)F=F2(DR)
       IF(DR.GE.2.0)F=F3(DR)
       CNDV=CNDV+F/C
90     X=TRANSFER(NSLMAX+NL+I)*PROPRT(I,4)**0.333333
       Y=X*CNDV
       Z(NL+I)=0.0
```

```
      RETURN
C
C     THERMAL CONDUCTIVITY OF THE LIQUID PHASE
C     ROBBINS AND KINGREA'S METHOD
C
120   DO 130 I=1,NOCOMP
      TRANSFER(NSLMAX+NL+I)=Z(NL+I)
130   Z(NL+I)=0.0
      CMOLE=0.0
      DO 160 I=1,NOCOMP
      IF(TRANSFER(NSLMAX+NL+I).LT.0.001)GO TO 160
      Z(NL+I)=1.0
      TR5=1.0
      D1=DENSTY(Z)
      D2=D1/FACT5
      D=D2/PROPRT(I,4)
      CP=SPHEAT(Z)*FACT2*PROPRT(I,4)
      IF(D2.GT.1.0)GO TO 140
      TR5=0.55*PROPRT(I,6)/(Z(2)+TADD)
140   S=PRTEXTRA(I,29)*FACT2/PROPRT(I,8)+1.9872*ALOG(TZERO/PROPRT(I,8))
      CNDV=(88.0-4.94*PRTEXTRA(I,28))/1000./S*TR5*CP*D**1.33333
      IF(Z(3).LT.PRES1)GO TO 150
C *** PRESSURE EFFECT CORRECTION ON THE THERMAL CONDUCTIVITY
      DR=D1*PROPRT(I,7)/PROPRT(I,4)
      ZC=PROPRT(I,5)*PROPRT(I,7)/R/PROPRT(I,6)
      C=SQRT(PROPRT(I,4))*(PROPRT(I,6)/PROPRT(I,5)**4*FACT1)**0.1666666
      C=C*ZC**5
      IF(DR.LE.0.5)F=F1(DR)
      IF(DR.LT.2.0.AND.DR.GT.0.5)F=F2(DR)
      IF(DR.GE.2.0)F=F3(DR)
      CNDV=CNDV+F/C
150   X=CNDV*TRANSFER(NSLMAX+NL+I)
      CMOLE=CMOLE+X
      RNOM=RNOM+X*PROPRT(I,4)
      DENOM=DENOM+PROPRT(I,4)*TRANSFER(NSLMAX+NL+I)
      Z(NL+I)=0.0
160   CONTINUE
      CNDVTY=RNOM/DENOM
      IF(CMOLE.LT.CNDVTY)CNDVTY=CMOLE
170   CNDVTY=CNDVTY/FACT4
```

```
      RETURN
      END
      FUNCTION SURFTENS(TRANSFER)
C
C     SURFACE TENSION IS GIVEN IN (NEWTONS)/(METER)
C      OR  POUNDF PER FT
C
      DIMENSION X(30),EQCON(10),SAVE(15),TRANSFER(30)
      COMMON
     1/PROPS/PROPRT(10,25),US(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL,NW
     2/PROPE/PRTEXTRA(10,32)
C
C     CONVERSION FACTORS
C
      IF(IUNITS.GT.0)GO TO 10
      FACT1=62.4
      FACT2=6.85E-05
      PDIFF=500.
      TADD=460.
      GO TO 20
10    FACT1=1000.
      FACT2=1.0E-03
      PDIFF=35.0
      TADD=273.15
C
20    DO 30 I=1,NSLMAX
30    SAVE(I)=TRANSFER(I)
      T2=TRANSFER(2)
      CALL DEWBUB(1,TRANSFER,BP)
      IF(BP.GE.T2)GO TO 36
      WRITE(NW,35)
35    FORMAT(//10X,43H ** WARNING FROM SURFACE TENSION ROUTINE **  /
     *10X,50H MIXTURE TEMPERATURE HIGHER THAN ITS BUBBLE POINT  /
     *10X,36H BUBBLE POINT SET AS ITS TEMPERATURE  /)
      TRANSFER(2)=BP
36    SURFTENS=0.0
      DO 40 I=1,NL
40    X(I)=TRANSFER(I)
      X(5)=0.0
```

```
      DO 50 I=1,NOCOMP
      X(NL+I)=0.0
50    TRANSFER(NL+I)=0.0
      DO 60 I=1,NOCOMP
      TRANSFER(NL+I)=1.0
      X(NL+I)=1.0
      DX=DENSTY(X)
       DT=DENSTY(TRANSFER)
      STENS=(PRTEXTRA(I,30)/PROPRT(I,4)*(DX-DT)/FACT1)**4
      X(NL+I)=0.0
      TRANSFER(NL+I)=0.0
60    SURFTENS=SURFTENS+SAVE(NL+I)*STENS
      GO TO 120
C
65    SUM=0.0
      CALL KVALUE(1, TRANSFER,EQCON)
      DO 70 I=1,NOCOMP
      X(NL+I)=SAVE(NL+I)
70    SUM=SUM+TRANSFER(NL+I)
      DO 75 I=1,NOCOMP
75    TRANSFER(NL+I)=TRANSFER(NL+I)/SUM
80    DO 90 I=1,NOCOMP
      RML=RML+PROPRT(I,4)*X(NL+I)
90    RMV=RMV+PROPRT(I,4)*TRANSFER(NL+I)
      RML=DENSTY(X)/RML
      RMV=DENSTY(TRANSFER)/RMV
      PX=0.0
      PY=0.0
      DO 100   I=1,NOCOMP
      PX=PX+PRTEXTRA(I,30)*X(NL+I)
100   PY=PY+PRTEXTRA(I,30)*TRANSFER(NL+I)
      SURFTENS=((PX*RML-PY*RMV)/FACT1)**4
C
120   SURFTENS=SURFTENS*FACT2
      DO 130 I=1,NSLMAX
      TRANSFER(I)=SAVE(I)
130   TRANSFER(NSLMAX+I)=0.0
      TRANSFER(2)=T2
      RETURN
      END
```

Program ECHOER

```
      LIST (LP)
      PROGRAM (ECHOER)
      INPUT 1 = CR0
      OUTPUT 2= LP0
      COMPACT
      END
      MASTER ECHO
C
      DIMENSION
     *TITLE(10),BUFFER(10),VNAMES(30),A(20),
     *EEN(500),NEMAT(30,2),SMATRX(400),RNAME(30),VNAME(30),MODULE(30),
     *PROPRT(10,25),PPTEXTRA(10,35),USERPPT(10,10),SMODN(30),FORMS(2)
      DIMENSION
     *KPM(30,8),KSFLAG(30),KSEM(30,3),
     *NELIST(30,2),KPS(10),ENAME(30),SNAME(30),ERR(30)
C
      DATA ZERO/8H00000000/
C
      NR=1
      NWUSER=2
C
C     DATA READER
C
10    FORMAT(16I5)
20    FORMAT(10A8)
30    FORMAT(5E15.7)
      READ(NR,20)TITLE
      READ(NR,10)NEMAX,NSMAX,NOCOMP,NSLMAX,NPAR,NCOLPM,KONVRG,NPS,
     1ITRACE,NLOOPS,NLIST,NVNAME,NPROP,NOGO,IUNITS,NUSRPPT,NCTOT
     2,NPROPX,IRT,IERR
      READ(NR,30)TOL
      READ(NR,20)(SNAME(I),I=1,NSMAX)
      READ(NR,20)(ENAME(I),I=1,NEMAX)
      IF(IRT.EQ.0)GO TO 40
      READ(NR,20)(RNAME(I),I=1,NEMAX)
      READ(NR,10)(MODULE(I),I=1,NEMAX)
40    READ(NR,20)(VNAME(I),I=1,NVNAME)
      IF(NOCOMP.EQ.0)GO TO 55
      DO 50 I=1,NOCOMP
      READ(NR,20)(PROPRT(I,J),J=1,2)
      READ(NR,30)(PROPRT(I,J),J=3,NPROP)
```

```
60    READ(NR,30)(USERPPT(I,J),J=1,NUSRPPT)
65    READ(NR,10)((KPM(I,J),J=1,NCOLPM),I=1,NEMAX)
      READ(NR,10)((KSEM(I,J),J=1,3),I=1,NSMAX)
70    READ(NR,30)(SMATRX(I),I=1,NSLMAX*NSMAX)
      IF(NPAR.EQ.0)GO TO 75
      READ(NR,30)(EEN(I),I=1,NPAR)
      READ(NR,10)((NEMAT(I,J),J=1,2),I=1,NEMAX)
75    IF(NLIST.EQ.0)GO TO 80
      READ(NR,10)((NELIST(I,J),J=1,2),I=1,NLIST)
      READ(NR,10)(KSFLAG(I),I=1,NSMAX)
80    IF(NPS.EQ.0)GO TO 90
      READ(NR,10)(KPS(I),I=1,NPS)
90    IF(IERR.EQ.0)GO TO 100
      READ(NR,30)(ERR(I),I=1,NSMAX)
100   CONTINUE
C
C     DATA ECHO
C
800   WRITE(NWUSER,802)TITLE
802   FORMAT(1H1//1X,10A8/1X,80(1H*)/)
803   WRITE(NWUSER,804)
804   FORMAT(//10X,15H PROCESS MATRIX/11X,14(1H*)/)
      DO 820 I=1,NEMAX
      WRITE(NWUSER,806)ENAME(I)
806   FORMAT(/10X,13H UNIT NAME : ,A8)
      DO 812 J=2,NCOLPM
      JK=KPM(I,J)
      IF(JK)814,814,808
808   J1=J-1
      WRITE(NWUSER,810)J1,SNAME(JK)
810   FORMAT(15X,18H INPUT  STREAM NO.,I2,5X,A8)
812   CONTINUE
814   DO 819 K=J,NCOLPM
      JK=IABS(KPM(I,K))
      IF(JK)820,820,816
816   K1=K-J+1
      WRITE(NWUSER,818)K1,SNAME(JK)
818   FORMAT(15X,18H OUTPUT STREAM NO. ,I2,5X,A8)
819   CONTINUE
820   CONTINUE
```

```
817   WRITE(NWUSER,821)
821   FORMAT(//10X,14H PLANT FEEDS : /11X,13(1H*)/)
      DO 822 I=1,NSMAX
      IF(KSEM(I,2).EQ.0.AND.KSEM(I,3).NE.0)WRITE(NWUSER,827)SNAME(I)
822   CONTINUE
      WRITE(NWUSER,823)
823   FORMAT(//10X,16H PLANT OUTPUTS : /11X,15(1H*)/)
      DO 825 I=1,NSMAX
      IF(KSEM(I,3).EQ.0.AND.KSEM(I,2).NE.0)WRITE(NWUSER,827)SNAME(I)
825   CONTINUE
827   FORMAT(16X,A8)
      WRITE(NWUSER,824)
824   FORMAT(//10X,14H STREAM MATRIX/11X,13(1H*))
      NL=NSMAX/3
      NM=NSMAX-NL*3
      IF(NL)836,836,826
826   I1=-2
      DO 834 I=1,NL
      I1=I1+3
      I2=I1+2
      WRITE(NWUSER,828)(SNAME(K),K=I1,I2)
828   FORMAT(//32X,3(7X,A8)/)
      J1=-1
      DO 832 J=1,NSLMAX
      J1=J1+2
      WRITE(NWUSER,830)VNAME(J1),VNAME(J1+1),(SMATRX((L-1)*NSLMAX+J)
     .,L=I1,I2)
830   FORMAT(11X,2A8,4X,3F15.5)
832   CONTINUE
834   CONTINUE
836   IF(NM.EQ.0)GO TO 839
      NL=NL*3+1
      WRITE(NWUSER,828)(SNAME(K),K=NL,NSMAX)
      WRITE(NWUSER,837)
837   FORMAT(10X)
      J1=-1
      DO 838 J=1,NSLMAX
      J1=J1+2
838   WRITE(NWUSER,830)VNAME(J1),VNAME(J1+1),(SMATRX((L-1)*NSLMAX+J)
     .,L=NL,NSLMAX)
```

```
843   FORMAT(/10X,50H ** NO PARAMETERS HAVE BEEN GIVEN FOR THE UNITS **
     */)
      GO TO 847
841   DO 842 I=1,NEMAX
      J=NEMAT(I,1)
      K=NEMAT(I,2)
      WRITE(NWUSER,844)ENAME(I)
      IF(K.EQ.0)GO TO 842
      K=K+J-1
      WRITE(NWUSER,845)(EEN(L),L=J,K)
845   FORMAT(12X,5F13.4)
842   CONTINUE
844   FORMAT(/11X,A8)
847   WRITE(NWUSER,846)
846   FORMAT(//10X,23H UNITS AND MODELS NAMES/11X,22(1H*)/)
      IF(IRT.GT.0)GO TO 848
      WRITE(NWUSER,851)
851   FORMAT(/10X,40H ** NO ROUTINES NAMES HAVE BEEN GIVEN ** /)
      GO TO 853
848   DO 850 I=1,NEMAX
      CALL COMP8(RNAME(I),ZERO,ICOMP)
      IF(ICOMP.EQ.1)GO TO 849
      WRITE(NWUSER,852)ENAME(I),RNAME(I)
      GO TO 850
849   WRITE(NWUSER,8520)ENAME(I)
850   CONTINUE
8520  FORMAT(10X,' UNIT ',2X,A8,2X,'  HAS NO MODEL ASSIGNED TO IT')
852   FORMAT(10X,8H UNIT    ,A8,28H   REPRESENTED BY ROUTINE    ,A8)
853   WRITE(NWUSER,854)
854   FORMAT(//10X,19H PROBLEM DIMENSIONS/11X,18(1H*)/)
      WRITE(NWUSER,856)NEMAX,NSMAX,NOCOMP,NSLMAX
856   FORMAT(/10X,18H NUMBER OF UNITS = ,I8
     1/10X,20H NUMBER OF STREAMS = ,I6
     2/10X,24H NUMBER OF COMPONENTS = ,I2
     3/10X,17H STREAM LENGTH = ,I9)
      WRITE(NWUSER,858)
858   FORMAT(//10X,17H PROBLEM CONTROLS/11X,16(1H*)/)
      WRITE(NWUSER,860)NLOOPS,TOL
860   FORMAT(10X,34H MAXIMUM NUMBER OF LOOPS ALLOWED = ,I5/
     110X,20H TOLERANCE ALLOWED = ,F10.5)
```

```
      IF(KONVRG)862,862,866
862   WRITE(NWUSER,864)
864   FORMAT(10X,29H CONVERGENCE PROMOTER NOT SET)
      GO TO 870
866   WRITE(NWUSER,868)KONVRG
868   FORMAT(10X,25H CONVERGENCE PROMOTER NO.,I2,4H SET)
870   IF(ITRACE)876,872,873
872   WRITE(NWUSER,874)
      GO TO 880
873   WRITE(NWUSER,875)ITRACE
      GO TO 880
874   FORMAT(10X,23H TRACE SET ON ALL UNITS)
875   FORMAT(10X,17H TRACE SET EVERY  ,I3,6H LOOPS  )
876   WRITE(NWUSER,878)
878   FORMAT(10X,21H TRACE OPTION NOT SET)
880   IF(NLIST)882,882,886
882   WRITE(NWUSER,884)
884   FORMAT(//10X,28H CALCULATION ORDER NOT GIVEN/11X,27(1H*)/)
      GO TO 898
886   WRITE(NWUSER,888)
888   FORMAT(//10X,18H CALCULATION ORDER/11X,17(1H*))
      ISEQ=2-NELIST(1,2)
      DO 896 I=1,NLIST
      IS2=NELIST(I,2)/2
      IF(IS2-ISEQ)890,894,892
890   WRITE(NWUSER,891)
891   FORMAT(/12X,15H SEQUENTIAL SET/)
      ISEQ=0
      GO TO 894
892   WRITE(NWUSER,893)
893   FORMAT(/12X,13H RECYCLE LOOP/)
      ISEQ=1
894   K1=NELIST(I,1)
      WRITE(NWUSER,895)ENAME(K1)
895   FORMAT(11X,A8)
896   CONTINUE
898   WRITE(NWUSER,899)
899   FORMAT(///30X,17H END OF DATA ECHO  /31X,16(1H*)///)
C
      STOP
```

Program LIBPROP

```
      OVERLAY PROGRAM (PROP)
      OVERLAY(1,1) MAIN
      OVERLAY(1,2) VISC
      OVERLAY(1,3) SURFTENS
      OVERLAY(1,4) CNDVTY
      OVERLAY(2,1) HDATA
      OVERLAY(2,2) RKDATA,BOILPT,VAPRES
      OVERLAY(2,3) TOMOLE,TOMASS,TOBRIT,TOSI
      INPUT 1 = CR0
      INPUT 3 = CR1
      INPUT 4 = CR2
      OUTPUT 2= LP0
      USE 7 = /ARRAY
      COMPRESS INTEGER AND LOGICAL
      COMPACT
      TRACE 2,500
      END
      MASTER PROP
      DIMENSION TRANSFER(30)
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS,
     *NCTOT,NSLMAX,NL,NW
      X=0.0
      IGOTO=0
1     CALL MAIN(TRANSFER,X,IGOTO,IRL)
      IF(IGOTO.EQ.0)GO TO 100
      GO TO (10,20,30,40,50,60,70,80,90,100),IGOTO
10    CALL DEWBUB(IRL,TRANSFER,X)
      GO TO 1
20    X=DENSTY(TRANSFER)
      GO TO 1
30    CALL ENTHALPY(TRANSFER)
      GO TO 1
40    X=SURFTENS(TRANSFER)
      GO TO 1
50    X=VISC(TRANSFER)
      GO TO 1
60    X=SPHEAT(TRANSFER)
      GO TO 1
70    X=CNDVTY(TRANSFER)
      GO TO 1
```

```
80    CALL TEMP(TRANSFER)
      GO TO 1
90    CALL FLASH(TRANSFER,X)
      GO TO 1
100   STOP
      END
      SUBROUTINE MAIN(TRANSFER,X,IGOTO,IRL)
      DIMENSION A(20),BUFFER(10),VNAMES(18)
      DIMENSION TRANSFER(30)
      COMMON /PROPS/ PROPRT(10,25),UPPT(10,10),NOCOMP,IUNITS,NCTOT
     *,NSLMAX,NL,NW
      COMMON /PROPE/ PRTEXTRA(10,32)
      DATA VNAMES /
     *8HBRITISH ,8HS.I.     ,8HMIXED   ,8HSYSTEM  ,8HCOMPOSIT,
     *8HPROPERTY,8HEXIT     ,8HNEW     ,8HDEW     ,8HBUBBLE  ,
     *8HDENSITY ,8HENTHALPY,8HSURFACE ,8HVISCOSIT,8HSPECIFIC,
     *8HCONDUCTI,8HTEMPERAT,8HFLASH
     */
C
      IF(IGOTO.EQ.0)GO TO 10
      GO TO (1415,1580,1740,1894,2090,2240,2390,2535,2640),IGOTO
C
C
10    CALL DEFBUF(7,80,BUFFER)
C
      NL=5
      NR=1
      NW=2
      NRCOMP=3
      NRCOMPX=4
      IFRACT=0
      ISYST=0
      NSLMAX=5
C
C
      WRITE(NW,20)
20    FORMAT(/10X,'*** PHYSICAL PROPERTY PACKAGE MK. 1 ***'/)
C
C *** COMPONENTS NAMES NAMES
C
```

```
      READ(NR,4040)AN
      NOCOMP=AN
      IF(NOCOMP.EQ.0)GO TO 5900
      WRITE(NW,515)
515   FORMAT(2X,18HCOMPONENTS NAMES :)
520   READ(NR,4000)BUFFER
      N=-1
      CALL COPYER(16,A,BUFFER,N)
      J=-1
      N=N/2
      NC1=NC+N-1
      DO 525 I=NC,NC1
      J=J+2
      CALL COPY8(PROPRT(I,1),A(J))
525   CALL COPY8(PROPRT(I,2),A(J+1))
      NC=NC+N
      IF(NC.LE.NOCOMP)GO TO 520
      NC=0
530   PAUSE PROPT
      GO TO 535
535   READ(NRCOMP,4010)NCOMP
      READ(NRCOMP,4010)NPROP
      READ(NRCOMPX,4010)NCOMPX
      READ(NRCOMPX,4010)NPROPX
      DO 560 I=1,NCOMP
      READ(NRCOMP,4000)A(1),A(2)
      READ(NRCOMPX,4000)A(3)
      DO 540 J=1,NOCOMP
      CALL COMP8(A(1),PROPRT(J,1),ICOMP)
      IF(ICOMP.NE.1)GO TO 540
      CALL COMP8(A(2),PROPRT(J,2),ICOMP)
      IF(ICOMP.EQ.1)GO TO 545
540   CONTINUE
      GO TO 550
545   IF(PROPRT(J,3).GT.0.0)GO TO 550
      NC=NC+1
      READ(NRCOMP,4050)(PROPRT(J,M),M=3,NPROP)
      READ(NRCOMPX,4050)(PRTEXTRA(J,M),M=1,NPROPX)
      GO TO 559
550   DO 555 M=3,NPROP
555   READ(NRCOMP,4050)Z
```

```
      DO 556 M=1,NPROPX
556   READ(NRCOMPX,4050)ZZ
559   IF(NC.EQ.NOCOMP)GO TO 575
560   CONTINUE
      IF(NC.GE.NOCOMP)GO TO 575
      DO 570 I=1,NOCOMP
      IF(PROPRT(I,3).GT.0.0)GO TO 570
      WRITE(NW,565)PROPRT(I,1),PROPRT(I,2)
565   FORMAT(/2X,2A8,2X,43H IS NOT A RECOGNISED LIBRARY COMPONENT NAME/
     *2X,17H PLEASE CHANGE IT/)
      READ(NR,4000)BUFFER
      N=2
      CALL COPYER(16,A,BUFFER,N)
      CALL COPY8(PROPRT(I,1),A(1))
      CALL COPY8(PROPRT(I,2),A(2))
570   CONTINUE
      GO TO 530
575   NSLMAX=5+NOCOMP
      NCTOT=NOCOMP
C
C *** TYPE OF UNITS USED (BRITISH OR S.I.)
C
580   WRITE(NW,585)
585   FORMAT(2X,'WHICH TYPE OF UNITS ? (BRITISH OR S.I. OR MIXED)')
590   READ(NR,4000)BUFFER
      N=1
      CALL COPYER(8,A,BUFFER,N)
      DO 592 I=1,3
      CALL COMP8(A(1),VNAMES(I),ICOMP)
      IF(ICOMP.EQ.1)GO TO 596
592   CONTINUE
      WRITE(NW,595)
595   FORMAT(/2X,35H UNRECOGNISED ANSWER, PLEASE RETYPE/)
      GO TO 590
596   IUNIT2=I
      IUNITS=IUNIT2-1
      IF(IUNIT2.EQ.3)IUNITS=0
      IF(IUNITS.EQ.0)GO TO 1000
      DO 599 I=1,NOCOMP
      PROPRT(I,5)=PROPRT(I,5)/14.5
```

```
      IF(PRTEXTRA(I,1).EQ.0.0)PRTEXTRA(I,3)=(PRTEXTRA(I,3)-32.0)/1.8
      PRTEXTRA(I,8)=PRTEXTRA(I,8)/1.8
      PRTEXTRA(I,29)=PRTEXTRA(I,29)/1.8
      DO 599 J=14,19
      PROPRT(I,J)=PROPRT(I,J)*2.325
599   CONTINUE
C
C *** PROPERTY REQUIRED OR COMMAND TO CHANGE THE SYSTEM
C
1000  WRITE(NW,1010)
1010  FORMAT(/2X,'PROPERTY REQUIRED OR COMMAND')
1020  READ(NR,4000)BUFFER
      N=-1
      CALL COPYER(8,A,BUFFER,N)
      DO 1030 IPROP=7,18
      CALL COMP8(A(1),VNAMES(IPROP),ICOMP)
      IF(ICOMP.EQ.1)GO TO 1060
1030  CONTINUE
1040  WRITE(NW,1050)
1050  FORMAT(2X,'UNRECOGNISED PROPERTY OR COMMAND , PLEASE RETYPE')
      GO TO 1020
1060  GO TO (1040,1040,1040,1040,1040,
     *1040,5900,1100,1400,1400,
     *1540,1700,1850,2050,2200,
     *2350,2500,2600
     *),IPROP
C
C *** COMMAND TO CHANGE THE SYSTEM
C
1100  DO 1110 J=4,6
      CALL COMP8(A(2),VNAMES(J),ICOMP)
      I=J-3
      IF(ICOMP.EQ.1)GO TO 1120
1110  CONTINUE
      GO TO 1040
1120  ASSIGN 1000 TO KPROP
      IFRACT=0
      GO TO (1150,1200,1130),I
1130  ISYST=0
      GO TO 1020
```

```
1150  DO 1160 I=1,NOCOMP
1160  PROPRT(I,3)=0.0
      GO TO 500
C
C *** COMPONENTS MOLE FRACTIONS
C
1200  WRITE(NW,1210)
1210  FORMAT(2X,'SYSTEM COMPOSITION IN MOLE FRACTIONS'/
     *)
      TRANSFER(1)=100.
      DO 1230 I=1,NOCOMP
      WRITE(NW,1220)PROPRT(I,1),PROPRT(I,2)
1220  FORMAT(2X,2A8)
      READ(NR,4040)TRANSFER(NL+I)
1230  CONTINUE
      IFRACT=1
      SUM=0.
      DO 1240 I=1,NOCOMP
1240  SUM=SUM+TRANSFER(NL+I)
      DO 1250 I=1,NOCOMP
1250  TRANSFER(NL+I)=TRANSFER(NL+I)/SUM
      GO TO KPROP
C
C *** DEW POINT - BUBBLE POINT
C
1400  IF(IFRACT.GT.0)GO TO 1405
      ASSIGN 1405 TO KPROP
      GO TO 1200
1405  ASSIGN 1410 TO KPRES
      GO TO 5000
1410  IRL=IPROP-9
      IGOTO=1
      RETURN
1415  TF=X
      IF(IUNIT2.NE.3)GO TO 1416
      TF=(TF-32.)/1.8
1416  IF(IPROP.EQ.10)GO TO 1460
      GO TO(1420,1450,1450),IUNIT2
1420  WRITE(NW,1430)TF
1430  FORMAT(2X,'DEW POINT = ',F10.3,' DEGREES F')
```

```
1470  GO TO (1480,1510,1510),IUNIT2
1480  WRITE(NW,1490)TF
1490  FORMAT(2X,'BUBBLE POINT = ',F10.3,' DEGREES F')
      GO TO 1000
1510  WRITE(NW,1520)TF
1520  FORMAT(2X,'BUBBLE POINT = ',F10.3,' DEGREES C')
      GO TO 1000
C
C *** DENSITY
C
1540  IF(IFRACT.GT.0)GO TO 1550
      ASSIGN 1550 TO KPROP
      GO TO 1200
1550  ASSIGN 1560 TO KPRES
      GO TO 5000
1560  ASSIGN 1570 TO KTEMP
      GO TO 5100
1570  WRITE(NW,5210)
      READ(NR,4040)TRANSFER(5)
      IGOTO=2
      RETURN
1580  D=X
1590  GO TO(1600,1620,1640),IUNIT2
1600  WRITE(NW,1610)D
1610  FORMAT(2X,'DENSITY = ',F10.4,' LBM/CU.FT.')
      GO TO 1000
1620  WRITE(NW,1630)D
1630  FORMAT(2X,'DENSITY = ',F10.4,' KGM/CU.METER')
      GO TO 1000
1640  D=D/62.4
      WRITE(NW,1650)D
1650  FORMAT(2X,'DENSITY = ',F10.4,' GM/CC')
      GO TO 1000
C
C *** ENTHALPY
C
1700  IF(IFRACT.GT.0)GO TO 1710
      ASSIGN 1710 TO KPROP
      GO TO 1200
1710  ASSIGN 1720 TO KPRES
```

```
      GO TO 5000
1720  ASSIGN 1730 TO KTEMP
      GO TO 5100
1730  WRITE(NW,5210)
      READ(NR,4040)TRANSFER(5)
      IGOTO=3
      RETURN
1740  GO TO(1750,1770,1790),IUNIT2
1750  WRITE(NW,1760)TRANSFER(4)
1760  FORMAT(2X,'ENTHALPY = ',F10.1,' BTU/LBMOLE')
      GO TO 1000
1770  WRITE(NW,1780)TRANSFER(4)
1780  FORMAT(2X,'ENTHALPY = ',F10.1,' KJOULE/KGMOLE')
      GO TO 1000
1790  TRANSFER(4)=TRANSFER(4)/1.8
      WRITE(NW,1800)TRANSFER(4)
1800  FORMAT(2X,'ENTHALPY = ',F10.1,' CALS/GMOLE')
      GO TO 1000
C
C *** SURFACE TENSION
C
1850  IF(IFRACT.GT.0)GO TO 1860
      ASSIGN 1860 TO KPROP
      GO TO 1200
1860  ASSIGN 1880 TO KPRES
      GO TO 5000
1880  ASSIGN 1890 TO KTEMP
      GO TO 5100
1890  IGOTO=4
      RETURN
1894  ST=X
1895  GO TO (1900,1920,1940),IUNIT2
1900  WRITE(NW,1910)ST
1910  FORMAT(2X,'SURFACE TENSION = ',1E10.6,' LBF/FT')
      GO TO 1000
1920  WRITE(NW,1930)ST
1930  FORMAT(2X,'SURFACE TENSION = ',1E10.6,' NEWTONS/METER')
      GO TO 1000
1940  ST=ST*14700.
      WRITE(NW,1950)ST
```

```
C
2050  IF(IFRACT.GT.0)GO TO 2060
      ASSIGN 2060 TO KPROP
      GO TO 1200
2060  ASSIGN 2070 TO KPRES
      GO TO 5000
2070  ASSIGN 2080 TO KTEMP
      GO TO 5100
2080  WRITE(NW,5210)
      READ(NR,4040)TRANSFER(5)
      IGOTO=5
      RETURN
2090  VS=X
      GO TO (2100,2120,2140),IUNIT2
2100  WRITE(NW,2110)VS
2110  FORMAT(2X,'VISCOSITY = ',1E10.6,' LB/FT SEC')
      GO TO 1000
2120  WRITE(NW,2130)VS
2130  FORMAT(2X,'VISCOSITY = ',1E10.6,' KGM/M SEC')
      GO TO 1000
2140  VS=VS*1488.16
      WRITE(NW,2150)VS
2150  FORMAT(2X,'VISCOSITY = ',1F10.4,' CENTIPOISES')
      GO TO 1000
C
C *** SPECIFIC HEAT
C
2200  IF(IFRACT.GT.0)GO TO 2210
      ASSIGN 2210 TO KPROP
      GO TO 1200
2210  ASSIGN 2220 TO KPRES
      GO TO 5000
2220  ASSIGN 2230 TO KTEMP
      GO TO 5100
2230  WRITE(NW,5210)
      READ(NR,4040)TRANSFER(5)
      IGOTO=6
      RETURN
2240  SP=X
      GO TO(2250,2270,2290),IUNIT2
```

```
2250  WRITE(NW,2260)SP
2260  FORMAT(2X,'SPECIFIC HEAT = ',1F10.4,' BTU/LB DEG.F')
      GO TO 1000
2270  WRITE(NW,2280)SP
2280  FORMAT(2X,'SPECIFIC HEAT = ',1E10.6,' KJOULES/KGM DEG.C')
      GO TO 1000
2290  WRITE(NW,2300)SP
2300  FORMAT(2X,'SPECIFIC HEAT = ',1F10.4,' CALS/GM DEG.C')
      GO TO 1000
C
C *** THERMAL CONDUCTIVITY
C
2350  IF(IFRACT.GT.0)GO TO 2360
      ASSIGN 2360 TO KPROP
      GO TO 1200
2360  ASSIGN 2370 TO KPRES
      GO TO 5000
2370  ASSIGN 2380 TO KTEMP
      GO TO 5100
2380  WRITE(NW,5210)
      READ(NR,4040)TRANSFER(5)
      IGOTO=7
      RETURN
2390  TC=X
      GO TO (2400,2420,2440),IUNIT2
2400  WRITE(NW,2410)TC
2410  FORMAT(2X,'THERMAL CONDUCTIVITY = ',1E10.6,'BTU/FT SEC DEG.F')
      GO TO 1000
2420  WRITE(NW,2430)TC
2430  FORMAT(2X,'THERMAL CONDUCTIVITY = ',1E10.6,' KJOULES/M SEC DEG.C')
      GO TO 1000
2440  TC=TC*15.12
      WRITE(NW,2450)TC
2450  FORMAT(2X,'THERMAL CONDUCTIVITY = ',1E10.6,' CALS/CM SEC DEG.C')
      GO TO 1000
C
C *** TEMPERATURE
C
2500  IF(IFRACT.GT.0)GO TO 2510
      ASSIGN 2510 TO KPROP
```

```
      GO TO 5220
2530  WRITE(NW,5210)
      READ(NR,4040)TRANSFER(5)
      IGOTO=8
      RETURN
2535  GO TO(2540,2570,2560),IUNIT2
2540  WRITE(NW,2550)TRANSFER(2)
2550  FORMAT(2X,'TEMPERATURE = ',F10.2,' DEGREES F')
      GO TO 1000
2560  TRANSFER(2)=(TRANSFER(2)-32.)/1.8
2570  WRITE(NW,2580)TRANSFER(2)
2580  FORMAT(2X,'TEMPERATURE = ',F10.2,' DEGREES C')
      IF(IUNIT2.EQ.3)TRANSFER(2)=TRANSFER(2)*1.8+32.
      GO TO 1000
C
C *** FLASH
C
2600  IF(IFRACT.GT.0)GO TO 2610
      ASSIGN 2610 TO KPROP
      GOTO 1200
2610  ASSIGN 2620 TO KPRES
      GOTO 5000
2620  ASSIGN 2630 TO KTEMP
      GO TO 5100
2630  X=TRANSFER(3)
      IGOTO=9
      RETURN
2640  IFRACT=0
2660  WRITE(NW,2670)
2670  FORMAT(2X,'PHASES COMPOSITIONS'/23X,'VAPOUR',4X,'LIQUID')
      WRITE(NW,2675)TRANSFER(1),TRANSFER(NSLMAX+1)
2675  FORMAT(2X,'FLOW PERCENTAGES',2F10.2)
      DO 2690 I=1,NOCOMP
      WRITE(NW,2680)PROPRT(I,1),PROPRT(I,2),TRANSFER(NL+I),
     +TRANSFER(NSLMAX+NL+I)
2680  FORMAT(2X,2A8,2X,2F10.6)
2690  CONTINUE
      GO TO 1000
C
C *** ALPHANUMERIC-NUMERIC READ-WRITE FORMATS
```

```
C
4000  FORMAT(10A8)
4010  FORMAT(16I5)
4040  FORMAT(10F0.0)
4050  FORMAT(1E15.7)
C
C *** FORMATS TO READ THE PRESSURE AND THE TEMPERATURE
C
5000  GO TO (5010,5030,5050),IUNIT2
5010  WRITE(NW,5020)
5020  FORMAT(2X,'SYSTEM PRESSURE IN P.S.I.A.')
      GO TO 5070
5030  WRITE(NW,5040)
5040  FORMAT(2X,'SYSTEM PRESSURE IN BARS')
      GO TO 5070
5050  WRITE(NW,5060)
5060  FORMAT(2X,'SYSTEM PRESSURE IN ATMS')
5070  READ(NR,4040)TRANSFER(3)
      IF(IUNIT2.EQ.3)TRANSFER(3)=TRANSFER(3)*14.69
      TRANSFER(NSLMAX+3)=TRANSFER(3)
      GO TO KPRES
5100  GO TO (5110,5130,5130),IUNIT2
5110  WRITE(NW,5120)
5120  FORMAT(2X,'SYSTEM TEMPERATURE IN DEGREES F')
      GO TO 5150
5130  WRITE(NW,5140)
5140  FORMAT(2X,'SYSTEM TEMPERATURE IN DEGREES C')
5150  READ(NR,4040)TRANSFER(2)
      IF(IUNIT2.EQ.3)TRANSFER(2)=TRANSFER(2)*1.8+32.
      TRANSFER(NSLMAX+2)=TRANSFER(2)
      GO TO KTEMP
5210  FORMAT(2X,'SYSTEM VAPOUR FRACTION')
5220  GO TO (5230,5250,5270),IUNITS
5230  WRITE(NW,5240)
5240  FORMAT(2X,'SYSTEM ENTHALPY IN BTU/LBMOLE')
      GO TO 5290
5250  WRITE(NW,5260)
5260  FORMAT(2X,'SYSTEM ENTHALPY IN KJOULES/KGMOLE')
      GO TO 5290
5270  WRITE(NW,5280)
```

```
C
5900  WRITE(NW,5910)
5910  FORMAT(/10X,'*** PHYSICAL PROPERTY PACKAGE CLOSED ***')
6000  STOP
      END
C
      FUNCTION PUREVF(I,TRANSFER)
C
      DIMENSION TRANSFER(30)
C
      TIN=TRANSFER(2)
      HIN=TRANSFER(4)
      BP=BOILPT(I,TRANSFER(3))
      TRANSFER(2)=BP
      TRANSFER(5)=3.0
      CALL ENTHALPY(TRANSFER)
      HV=TRANSFER(4)
      TRANSFER(5)=2.0
      CALL ENTHALPY(TRANSFER)
      HL=TRANSFER(4)
      PUREVF=(HIN-HL)/(HV-HL)
      IF(PUREVF.LT.0.0)PUREVF=0.0
      IF(PUREVF.GT.1.0)PUREVF=1.0
      TRANSFER(5)=PUREVF
      TRANSFER(2)=TIN
      TRANSFER(4)=HIN
      RETURN
      END
      SUBROUTINE COPYER(M,A,B,NCOPY)
C
      DIMENSION FORMS(2),FORM(3)
      DIMENSION A(20),B(10),SP(2)
      DATA FORM/8H(        ,8H        ,8HX,1F0.0)/
      DATA SP/8H        ,8H        /
C
      IF(NCOPY+1)1,2,3
1     NC=-1
      NCOPY=-NCOPY-1
      GO TO 5
2     NCOPY=81
```

```
3       NC=0
5       K=1
        N=1
        DO 60 I=1,80
        J=1
        CALL COMP(J,B(1),I,SP(1),1)
        IF(J)10,10,20
10      CALL COPY(1,A(N),K,B(1),I)
        IF(K.GE.M)GO TO 50
        K=K+1
        GO TO 60
20      IF(K-1)60,60,30
30      L=M-K+1
        IF(L)50,50,40
40      CALL COPY(L,A(N),K,SP(1),1)
50      K=1
        N=N+M/8
        IF(N-NCOPY)60,60,70
60      CONTINUE
70      IF(NC.GE.0)GO TO 110
        I=I-1
        WRITE(8,100)I
100     FORMAT(I8)
        READ(8,101)FORM(2)
101     FORMAT(A8)
        READ(7,FORM)A(NCOPY+1)
110     NCOPY=N-1
        RETURN
        END
        FINISH
```

MACROS

PEETREADER

PEETUPDATER

BATCHREAD

BATCHUPDATE

INDPEETREAD

INDPEETUP

PROPACK

MACRO PEETREADER

```
   TA NONE
   IF STRING(%A)=(),GO TO 8
   IF NOT EXI(PEETLOG),GO TO 1B
   LF PEETLOG
   ER PEETLOG
1B MAXSIZE 24000
   LOAD INTERACTBIN
1  ONLINE *CR0
   ONLINE *LP0
   ASSIGN *CR1,PROPTS
   ASSIGN *CR4,PPTEXTRA
   IF EXI(%A),ASSIGN *CR2,%A
   IF NOT STRING(%B)=(),ASSIGN *CR3,%B
   IF NOT STRING(%C)=(),ASSIGN *CR5,%C
   IF NOT STRING(%D)=(),ASSIGN *CR6,%D
2  ENTER 0
3  IF NOT HALTED(FILEA),GO TO 4
   RELEASE *CR2
   IF EXI(%A),ER %A
   ASSIGN *CP0,%A
   RESUME
   GO TO 3
4  IF NOT HALTED(PROPT),GO TO 5
   RELEASE *CR1
   RELEASE *CR4
   ASSIGN *CR1,PROPTS
   ASSIGN *CR4,PPTEXTRA
   IF STRING(%C)=(),GO TO 4A
   RELEASE *CR5
   ASSIGN *CR5,%C
4A IF STRING(%D)=(),GO TO 4B
   RELEASE *CR6
   ASSIGN *CR6,%D
4B RESUME
   GO TO 3
5  IF NOT HALTED(USRPT),GO TO 6
   RELEASE *CR3
   IF STRING(%B)=(),GO TO 9
   ASSIGN *CR3,%B
   RESUME
   GO TO 3
```

```
7  IF NOT EXI(%A), GO TO 8
   DELETE INTERACTBIN
   LOAD ECHOERBIN
   ASSIGN *CR0,%A
   ASSIGN *LP0,PEETLOG(APPEND)
   ENTER 0
   LF PEETLOG,*LP
EXIT
8  DP 0,YOU HAVE NOT ASSIGNED A DUMP
   DP 0,FILE NAME.
   DP 0,PLEASE DO SO AFTER THE MACRO
   DP 0,NAME  AND RECALL THE MACRO.
EXIT
9  DP 0,ERROR IN LIBRARY COMPONENT NAMES GIVEN
   DP 0,EITHER YOU HAVE GIVEN WRONG NAMES,
   DP 0,OR YOU FORGOT TO ASSIGN AN
   DP 0,ADDITIONAL DATA FILE NAME.
   DP 0,PLEASE CORRECT YOUR ERROR
   DP 0,AND RE-ENTER THROUGH PEETUPDATER
EXIT
10 IF HALTED(ER),GO TO 7
   DELETE INTERACTBIN
   LOAD PEETPACKBIN
   ASSIGN *CR0,%A
   ONLINE *LP0
   ASSIGN *LP1,PEETLOG(APPEND)
   ENTER 0
   IF NOT HALTED(FILEA),GO TO 11
   RELEASE *CR0
   ERASE %A
   ASSIGN *CP0,%A
   RESUME
11 IF HALTED(ER),GO TO 7
   DELETE PEETPACKBIN
   LOAD INTERACTBIN
   ON 1
   GO TO 1
```

MACRO PEETUPDATER

```
   TA NONE
   IF STRING(%A)=(),GO TO 8
   IF NOT EXI(PEETLOG),GO TO 1B
   LF PEETLOG,*LP
   ER PEETLOG
1B MAXSIZE 24000
   LOAD INTERACTBIN
   ON 1
1  ONLINE *CR0
   ONLINE *LP0
   ASSIGN *CR1,PROPTS
   ASSIGN *CR4,PPTEXTRA
   IF EXI(%A),ASSIGN *CR2,%A
   IF NOT STRING(%B)=(),ASSIGN *CR3,%B
2  ENTER 0
3  IF NOT HALTED(FILEA),GO TO 4
   RELEASE *CR2
   IF EXI(%A),ER %A
   ASSIGN *CP0,%A
   RESUME
   GO TO 3
4  IF NOT HALTED(PROPT),GO TO 5
   RELEASE *CR1
   RELEASE *CR4
   ASSIGN *CR1,PROPTS
   ASSIGN *CR4,PPTEXTRA
4B RESUME
   GO TO 3
5  IF NOT HALTED(USRPT),GO TO 6
   RELEASE *CR3
   IF STRING(%B)=(),GO TO 9
   ASSIGN *CR3,%B
   RESUME
   GO TO 3
6  IF NOT HALTED(EXIT), GO TO 10
7  IF NOT EXI(%A), GO TO 8
   DELETE INTERACTBIN
   LOAD ECHOERBIN
   ASSIGN *CR0,%A
   ASSIGN *LP0,PEETLOG(APPEND)
   ENTER 0
```

```
EXIT
#  TA AB,CM
8  DP 0,YOU HAVE NOT ASSIGNED A DUMP
   DP 0,FILE NAME.
   DP 0,PLEASE DO SO AFTER THE MACRO
   DP 0,NAME  AND RECALL THE MACRO.
#  TA NONE
EXIT
#  TA AB,CM
9  DP 0,ERROR IN LIBRARY COMPONENT NAMES GIVEN
   DP 0,EITHER YOU HAVE GIVEN WRONG NAMES,
   DP 0,OR YOU FORGOT TO ASSIGN AN
   DP 0,ADDITIONAL DATA FILE NAME.
   DP 0,PLEASE CORRECT YOUR ERROR
   DP 0,AND RE-ENTER THROUGH PEETUPDATER
#  TA NONE
EXIT
10 IF HALTED(ER),GO TO 7
   DELETE INTERACTBIN
   MAXSIZE 60000
12 LOAD PEETPACKBIN
13 ASSIGN *CR0,%A
   ONLINE *LP1
   ASSIGN *LP0,PEETLOG(APPEND)
   ENTER 0
   IF NOT HALTED(FILEA),GO TO 14
   RELEASE *CR0
   ERASE %A
   ASSIGN *CP0,%A
   RESUME
14 IF HALTED(ER),GO TO 7
   DELETE PEETPACKBIN
   LOAD INTERACTBIN
   ON 1
   GO TO 1
```

MACRO BATCHREAD

```
#   TA NONE
    IF STRING(%A)=(),GO TO 7
    IF NOT EXI(PEETLOG),GO TO 1
    LF PEETLOG,*LP
    ER PEETLOG
1   MAXSIZE 24000
    LOAD BATCHBIN
    ONLINE *CR0
    ASSIGN *LP0,PEETLOG
    ASSIGN *CR1,PROPTS
    ASSIGN *CR5,PPTEXTRA
    IF NOT STRING(%B)=(),ASSIGN *CR3,%B
2   IF EXI(%A),ASSIGN *CR2,%A
    ENTER 0
3   IF NOT HALTED(FILEA),GO TO 3A
    RELEASE *CR2
    IF EXI(%A),ER %A
    ASSIGN *CP0,%A
    ONLINE *CR2
    RESUME
    GO TO 3
3A  IF HALTED(FILEB),GO TO 4
    IF HALTED(FILEC),GO TO 5
    IF HALTED(EXIT),GO TO 6
    IF HALTED(ER),GO TO 6
    IF HALTED(EE),EXIT
    GO TO 10
4   IF STRING(%B)=(),GO TO 8
    RELEASE *CR3
    ASSIGN *CR3,%B
    IF STRING(%D)=(),GO TO 4A
    RELEASE *CR6
    ASSIGN *CR6,%D
4A  RESUME
    GO TO 3
5   IF STRING(%C)=(),GO TO 9
    RELEASE *CR4
    ASSIGN *CR4,%C
    RESUME
    GO TO 3
6   DELETE BATCHBIN
```

```
    ASSIGN *CR0,%A
    ASSIGN *LP0,PEETLOG(APPEND)
    ENTER 0
    LF PEETLOG,*LP
EXIT
EJ
#   TA AB,CM
7   DISPLAY 0,YOU HAVE FORGOTTEN TO ASSIGN THE
    DISPLAY 0,FILE IN WHICH THE DATA SHOULD BE
    DISPLAY 0,STORED FOR SUBSEQUENT USE.
    DISPLAY 0,PLEASE DO SO AFTER THE MACRO NAME.
#   TA NONE
EXIT
EJ
#   TA AB,CM
8   DISPLAY 0,ERROR IN LIBRARY COMPONENT NAMES
    DISPLAY 0,GIVEN.
    DISPLAY 0,PLEASE CORRECT THE WRONG ONES
    DISPLAY 0,AND RE-INPUT ALL THE DATA.
#   TA NONE
    ERASE %A
EXIT
EJ
#   TA AB,CM
9   DISPLAY 0,YOU HAVE FORGOTTEN TO GIVE YOUR
    DISPLAY 0,OWN COMPONENT DATA FILE.
    DISPLAY 0,PLEASE DO SO AND RE-ENTER ALL
    DISPLAY 0,YOUR DATA.
#   TA NONE
EXIT
EJ
10  DELETE BATCHBIN
    MAXSIZE 60000
    SP F,(%ABIN)
    IF EXI(%F),GO TO 11
    IF ABS(LIB),GO TO 12
    UAFORTRAN PROG NORMANSPACK,OWNPD,LIB :ECP0750.EXECUTIVESUBS,-
    LIB %(LIB),LIB :ECP0750.MODELSUBS,LIB :ECP0750.PHYSUBS,-
    LIB :ECP0750.THERMOSUBS,SAVE %F,NORUN,EXIT
11  LOAD %F
    GO TO 13
```

```
13 ASSIGN *CR0,%A
   ASSIGN *LP0,PEETLOG(APPEND)
   ASSIGN *LP1,PEETLOG(APPEND)
   ENTER 0
14 IF HALTED(FILEA),GO TO 15
   IF HALTED(ER),GO TO 6
   IF HALTED(EE),EXIT
   GO TO 20
15 RELEASE *CR0
   ERASE %A
   ASSIGN *CP0,%A
   ONLINE *CR0
   RESUME
   GO TO 14
20 DELETE PEETPACKBIN
   LOAD BATCHBIN
   ON 1
   ONLINE *CR0
   ASSIGN *CR1,PROPTS
   ASSIGN *CR2,%A
   ASSIGN *CR5,PPTEXTRA
   IF NOT STRING(%B)=(),ASSIGN *CR3,%B
21 ASSIGN *LP0,PEETLOG(APPEND)
   ENTER 0
   GO TO 3
```

MACRO BATCHUPDATE

```
#  TA NONE
   IF STRING(%A)=(),GO TO 7
   IF NOT EXI(PEETLOG),GO TO 1
   LF PEETLOG,*LP
   ER PEETLOG
1  MAXSIZE 24000
   LOAD BATCHBIN
   ON 1
   ONLINE *CR0
   ASSIGN *LP0,PEETLOG
   ASSIGN *CR1,PROPTS
   ASSIGN *CR5,PPTEXTRA
   IF NOT STRING(%B)=(),ASSIGN *CR3,%B
2  IF EXI(%A),ASSIGN *CR2,%A
   ENTER 0
3  IF NOT HALTED(FILEA),GO TO 3A
   RELEASE *CR2
   IF EXI(%A),ER %A
   ASSIGN *CP0,%A
   ONLINE *CR2
   RESUME
   GO TO 3
3A IF HALTED(FILEB),GO TO 4
   IF HALTED(FILEC),GO TO 5
   IF HALTED(EXIT),GO TO 6
   IF HALTED(ER),GO TO 6
   IF HALTED(EE),EXIT
   GO TO 10
4  IF STRING(%B)=(),GO TO 8
   RELEASE *CR3
   ASSIGN *CR3,%B
   IF STRING(%D)=(),GO TO 4A
   RELEASE *CR6
   ASSIGN *CR6,%D
4A RESUME
   GO TO 3
5  IF STRING(%C)=(),GO TO 9
   RELEASE *CR4
   ASSIGN *CR4,%C
   RESUME
   GO TO 3
```

```
    LOAD ECHOERBIN
    ASSIGN *CR0,%A
    ASSIGN *LP0,PEETLOG(APPEND)
    ENTER 0
    LF PEETLOG,*LP
EXIT
EJ
#   TA AB,CM
7   DISPLAY 0,YOU HAVE FORGOTTEN TO ASSIGN THE
    DISPLAY 0,FILE IN WHICH THE DATA SHOULD BE
    DISPLAY 0,STORED FOR SUBSEQUENT USE.
    DISPLAY 0,PLEASE DO SO AFTER THE MACRO NAME.
#   TA NONE
EXIT
EJ
#   TA AB,CM
8   DISPLAY 0,ERROR IN LIBRARY COMPONENT NAMES
    DISPLAY 0,GIVEN.
    DISPLAY 0,PLEASE CORRECT THE WRONG ONES
    DISPLAY 0,AND RE-INPUT ALL THE DATA.
#   TA NONE
    ERASE %A
EXIT
EJ
#   TA AB,CM
9   DISPLAY 0,YOU HAVE FORGOTTEN TO GIVE YOUR
    DISPLAY 0,OWN COMPONENT DATA FILE.
    DISPLAY 0,PLEASE DO SO AND RE-ENTER ALL
    DISPLAY 0,YOUR DATA.
#   TA NONE
EXIT
EJ
10  DELETE BATCHBIN
    MAXSIZE 60000
    SP F,(%ABIN)
    IF EXI(%F),GO TO 11
    IF ABS(LIB),GO TO 12
    UAFORTRAN PROG NORMANSPACK,OWNPD,LIB :ECP0750.EXECUTIVESUBS,-
    LIB %(LIB),LIB :ECP0750.MODELSUBS,LIB :ECP0750.PHYSUBS,-
    LIB :ECP0750.THERMOSUBS,SAVE %F,NORUN,EXIT
11  LOAD %F
```

```
12 LOAD PEETPACKBIN
13 ASSIGN *CR0,%A
   ASSIGN *LP0,PEETLOG(APPEND)
   ASSIGN *LP1,PEETLOG(APPEND)
   ENTER 0
14 IF HALTED(FILEA),GO TO 15
   IF HALTED(ER),GO TO 6
   IF HALTED(EE),EXIT
   GO TO 20
15 RELEASE *CR0
   ERASE %A
   ASSIGN *CP0,%A
   ONLINE *CR0
   RESUME
   GO TO 14
20 DELETE PEETPACKBIN
   LOAD BATCHBIN
   ON 1
   ONLINE *CR0
   ASSIGN *CR1,PROPTS
   ASSIGN *CR2,%A
   ASSIGN *CR5,PPTEXTRA
   IF NOT STRING(%B)=(),ASSIGN *CR3,%B
21 ASSIGN *LP0,PEETLOG(APPEND)
   ENTER 0
   GO TO 3
```

MACRO INDPEETREAD

```
#  TA NONE
   IF STRING(%A)=(),GO TO 20
   RV PEETPACK
   IF NOT EXI(PEETLOG),GO TO 1B
   LF PEETLOG
   ER PEETLOG
1B MAXSIZE 24000
   LOAD INTERACTBIN
   ON 2
   ONLINE *CR0
   ONLINE *LP0
   ASSIGN *CR1,PROPTS
   ASSIGN *CR4,PPTEXTRA
   IF EXI(%A),ASSIGN *CR2,%A
   IF NOT STRING(%B)=(),ASSIGN *CR3,%B
   IF NOT STRING(%C)=(),ASSIGN *CR5,%C
   IF NOT STRING(%D)=(),ASSIGN *CR6,%D
2  ENTER 0
3  IF NOT HALTED(FILEA),GO TO 4
   RELEASE *CR2
   IF EXI(%A),ER %A
   ASSIGN *CP0,%A
   RESUME
   GO TO 3
4  IF NOT HALTED(PROPT),GO TO 5
   RELEASE *CR1
   RELEASE *CR4
   ASSIGN *CR1,PROPTS
   ASSIGN *CR4,PPTEXTRA
   IF STRING(%C)=(),GO TO 4A
   RELEASE *CR5
   ASSIGN *CR5,%C
4A IF STRING(%D)=(),GO TO 4B
   RELEASE *CR6
   ASSIGN *CR6,%D
4B RESUME
   GO TO 3
5  IF NOT HALTED(USRPT),GO TO 6
   RELEASE *CR3
   IF STRING(%B)=(),GO TO 9
   ASSIGN *CR3,%B
```

```
    GO TO 3
6   IF NOT HALTED(EXIT), GO TO 8
    DELETE INTERACTBIN
6A  LOAD ECHOERBIN
    ASSIGN *CR0,%A
    ASSIGN *LP0,PEETLOG(APPEND)
    ENTER 0
    LF PEETLOG,*LP
EXIT
8   IF NOT HALTED(EQCAL),GO TO 9
    RELEASE *CP0
    COPY PEETPACK,NPACK
    ASSIGN *CP0,NPACK(APPEND)
    RESUME
    GO TO 3
9   DELETE INTERACTBIN
    IF NOT HALTED (ER),GO TO 10
    IF NOT HALTED (EE), GO TO 10
    GO TO 6A
    EDIT NPACK
    TE
    IF       FINISH
    F
    E
    SP F,(%ABIN)
    IF PRE(LIB),GO TO 11
    UAFORTRAN PROG NPACK,OWNPD,LIB :ECP0750.EXECUTIVESUB,-
    LIB :ECP0750.MODELSUBS,LIB :ECP0750.PHYSUBS,-
    LIB :ECP0750.THERMOSUBS,-
    SAVE %F,NORUN,EXIT
    GO TO 12
11  IF EXI(%F),GO TO 12
    UAFORTRAN PROG NPACK,OWNPD,LIB :ECP0750.EXECUTIVESUB,-
    LIB %(LIB),LIB :ECP0750.MODELSUBS,LIB :ECP0750.PHYSUBS,-
    LIB :ECP0750.THERMOSUBS,-
    SAVE %F,NORUN,EXIT
12  LOAD %F
    ASSIGN *CR0,%A
    ONLINE *LP0
    ASSIGN *LP1,PEETLOG(APPEND)
    ENTER 0
```

```
    RELEASE *CRO
    ERASE %A
    ASSIGN *CPO,%A
    RESUME
13  IF HALTED(ER),GO TO 7
    DELETE %F
    LOAD INTERACTBIN
    ON 1
    ON 2
    GO TO 1
20  DP 0,YOU HAVE NOT ASSIGNED A DUMP
    DP 0,FILE NAME.
    DP 0,PLEASE DO SO AFTER THE MACRO
    DP 0,NAME  AND RECALL THE MACRO.
EXIT
21  DP 0,ERROR IN LIBRARY COMPONENT NAMES GIVEN
    DP 0,EITHER YOU HAVE GIVEN WRONG NAMES,
    DP 0,OR YOU FORGOT TO ASSIGN AN
    DP 0,ADDITIONAL DATA FILE NAME.
    DP 0,PLEASE CORRECT YOUR ERROR
    DP 0,AND RE-ENTER THROUGH PEETUPDATER
EXIT
```

MACRO IND PEETUP

```
#  TA NONE
   IF STRING(%A)=(),GO TO 20
   RV PEETPACK
   IF NOT EXI(PEETLOG),GO TO 1B
   LF PEETLOG
   ER PEETLOG
1B MAXSIZE 24000
   LOAD INTERACTBIN
   ON 2
   ON 1
   ONLINE *CR0
   ONLINE *LP0
   ASSIGN *CR1,PROPTS
   ASSIGN *CR4,PPTEXTRA
   IF EXI(%A),ASSIGN *CR2,%A
   IF NOT STRING(%B)=(),ASSIGN *CR3,%B
   IF NOT STRING(%C)=(),ASSIGN *CR5,%C
   IF NOT STRING(%D)=(),ASSIGN *CR6,%D
2  ENTER 0
3  IF NOT HALTED(FILEA),GO TO 4
   RELEASE *CR2
   IF EXI(%A),ER %A
   ASSIGN *CP0,%A
   RESUME
   GO TO 3
4  IF NOT HALTED(PROPT),GO TO 5
   RELEASE *CR1
   RELEASE *CR4
   ASSIGN *CR1,PROPTS
   ASSIGN *CR4,PPTEXTRA
   IF STRING(%C)=(),GO TO 4A
   RELEASE *CR5
   ASSIGN *CR5,%C
4A IF STRING(%D)=(),GO TO 4B
   RELEASE *CR6
   ASSIGN *CR6,%D
4B RESUME
   GO TO 3
5  IF NOT HALTED(USRPT),GO TO 6
   RELEASE *CR3
   IF STRING(%B)=(),GO TO [illegible]
```

```
    RESUME
    GO TO 3
6   IF NOT HALTED(EXIT), GO TO 8
    DELETE INTERACTBIN
6A  LOAD ECHOERBIN
    ASSIGN *CR0,%A
    ASSIGN *LP0,PEETLOG(APPEND)
    ENTER 0
    LF PEETLOG,*LP
EXIT
8   IF NOT HALTED(EQCAL),GO TO 9
    RELEASE *CP0
    COPY PEETPACK,NPACK
    ASSIGN *CP0,NPACK(APPEND)
    RESUME
    GO TO 3
9   DELETE INTERACTBIN
    IF NOT HALTED (ER),GO TO 10
    IF NOT HALTED (EE), GO TO 10
    GO TO 6A
    EDIT NPACK
    TE
    IF        FINISH
    F
    E
    SP F,(%ABIN)
    IF PRE(LIB),GO TO 11
    UAFORTRAN PROG NPACK,OWNPD,LIB :ECP0750.EXECUTIVESUB,-
    LIB :ECP0750.MODELSUBS,LIB :ECP0750.PHYSUBS,-
    LIB :ECP0750.THERMOSUBS,-
    SAVE %F,NORUN,EXIT
    GO TO 12
11  IF EXI(%F),GO TO 12
    UAFORTRAN PROG NPACK,OWNPD,LIB :ECP0750.EXECUTIVESUB,-
    LIB %(LIB),LIB :ECP0750.MODELSUBS,LIB :ECP0750.PHYSUBS,-
    LIB :ECP0750.THERMOSUBS,-
    SAVE %F,NORUN,EXIT
12  LOAD %F
    ASSIGN *CR0,%A
    ONLINE *LP0
    ASSIGN *LP1,PEETLOG(APPEND)
```

```
   IF NOT HALTED(FILEA),GO TO 13
   RELEASE *CRO
   ERASE %A
   ASSIGN *CPO,%A
   RESUME
13 IF HALTED(ER),GO TO 7
   DELETE %F
   LOAD INTERACTBIN
   ON 1
   ON 2
   GO TO 1
20 DP 0,YOU HAVE NOT ASSIGNED A DUMP
   DP 0,FILE NAME.
   DP 0,PLEASE DO SO AFTER THE MACRO
   DP 0,NAME  AND RECALL THE MACRO.
EXIT
21 DP 0,ERROR IN LIBRARY COMPONENT NAMES GIVEN
   DP 0,EITHER YOU HAVE GIVEN WRONG NAMES,
   DP 0,OR YOU FORGOT TO ASSIGN AN
   DP 0,ADDITIONAL DATA FILE NAME.
   DP 0,PLEASE CORRECT YOUR ERROR
   DP 0,AND RE-ENTER THROUGH PEETUPDATER
EXIT
```

MACRO PROPACK

```
   TA NONE
#  IF MOP,RP CF,OL,LS
   MAXSIZE 24000
   LOAD PROPBIN
   ONLINE *CR0
   ONLINE *LP0
   ASSIGN *CR1,PROPTS
   ASSIGN *CR2,PPTEXTRA
   ENTER 0
1  IF HALTED(PROPT),GO TO 2
2  RELEASE *CR1
   RELEASE *CR2
   ASSIGN *CR1,PROPTS
   ASSIGN *CR2,PPTEXTRA
   RESUME
   GO TO 1
```

Property Data File

PROPTS

```
  45
  21
HYDROGEN
  0.1000000E 01
  0.2000000E 01
  0.1908000E 03
  0.5990000E 02
  0.1040000E 01
  0.3730000E 02
  0.0000000E 00
  0.3250000E 01
  0.3100000E 02
  0.0000000E 00
  0.0000000E 00
  0.1576447E 04
  0.3357836E 01
  0.1273068E-03
  0.2656361E-07
 -0.4801999E-10
  0.1433455E-13
  0.2506000E 03
  0.5581837E 01
METHANE
  0.2000000E 01
  0.1600000E 02
  0.6731000E 03
  0.3439100E 03
  0.1590000E 01
  0.2013000E 03
  0.0000000E 00
  0.5680000E 01
  0.5200000E 02
  0.4278000E 00
  0.8670000E-01
  0.2278571E 03
  0.5170631E 00
  0.1316133E-03
  0.1413468E-06
 -0.8048506E-10
  0.1352522E-13
  0.2128800E 04
  0.7027729E 01
ETHANE
  0.3000000E 01
  0.3010000E 02
  0.7098000E 03
  0.5500100E 03
  0.2370000E 01
  0.3325000E 03
  0.1604000E 00
  0.6050000E 01
  0.6800000E 02
  0.4340000E 00
  0.8800000E-01
  0.1390360E 03
  0.3930300E 00
  0.2354850E-03
```

0.5419241E-07
-0.5226095E-10
0.1011986E-13
0.3739500E 04
0.7277361E 01
PROPANE
0.4000000E 01
0.4410000E 02
0.6174000E 03
0.6659500E 03
0.3210000E 01
0.4163000E 03
0.1538000E 00
0.6400000E 01
0.8400000E 02
0.4380000E 00
0.8890000E-01
0.1133416E 03
0.3677178E 00
0.2768229E-03
0.9134936E-08
-0.3367751E-10
0.7178678E-14
0.4811800E 04
0.7392262E 01
I-BUTANE
0.5000000E 01
0.5810000E 02
0.5291000E 03
0.7346500E 03
0.4210000E 01
0.4709000E 03
0.1825000E 00
0.6730000E 01
0.1055000E 03
0.4420000E 00
0.8980000E-01
0.1023728E 03
0.3634607E 00
0.3000314E-03
-0.2482609E-07
-0.1737302E-10
0.4377640E-14
0.5416200E 04
0.7349085E 01
N-BUTANE
0.6000000E 01
0.5810000E 02
0.5507000E 03
0.7653100E 03
0.4080000E 01
0.4911000E 03
0.1953000E 00
0.6730000E 01
0.1014000E 03
0.4450000E 00
0.9060000E-01

0.1136549E 03
0.3691676E 00
0.2740593E-03
0.1854880E-08
-0.2925181E-10
0.6371334E-14
0.5801200E 04
0.7492753E 01
I-PENTANE
0.7000000E 01
0.7220000E 02
0.4830000E 03
0.8298000E 03
0.4940000E 01
0.5421000E 03
0.2104000E 00
0.7020000E 01
0.1174000E 03
0.4450000E 00
0.9060000E-01
0.1025266E 03
0.3585026E 00
0.2936093E-03
-0.1601876E-07
-0.2229632E-10
0.5355312E-14
0.6470800E 04
0.5446800E 01
N-PENTANE
0.8000000E 01
0.7220000E 02
0.4895000E 03
0.8456000E 03
0.4980000E 01
0.5569000E 03
0.2387000E 00
0.7020000E 01
0.1161000E 03
0.4510000E 00
0.9190000E-01
0.1102691E 03
0.3660492E 00
0.2778974E-03
-0.7266540E-08
-0.2454261E-10
0.5523713E-14
0.6595100E 04
0.7489673E 01
NEO-PENTANE
0.9000000E 01
0.7220000E 02
0.4640000E 03
0.7807700E 03
0.4890000E 01
0.5091000E 03
0.1950000E 00
0.7020000E 01

```
  0.1233000E 03
  0.0000000E 00
  0.0000000E 00
  0.9526367E 02
  0.3640912E 00
  0.3238670E-03
 -0.4596739E-07
 -0.1111252E-10
  0.3856700E-14
  0.5648600E 04
  0.7263947E 01
N-HEXANE
  0.1000000E 02
  0.8620000E 02
  0.4400000E 03
  0.9142000E 03
  0.5900000E 01
  0.6157000E 03
  0.2927000E 00
  0.7270000E 01
  0.1316000E 03
  0.4590000E 00
  0.9350000E-01
  0.1087707E 03
  0.3630053E 00
  0.2782953E-03
 -0.7901680E-08
 -0.2500360E-10
  0.5705812E-14
  0.7627200E 04
  0.7717119E 01
N-HEPTANE
  0.1100000E 02
  0.1002000E 03
  0.3968000E 03
  0.9723100E 03
  0.6830000E 01
  0.6692000E 03
  0.3403000E 00
  0.7430000E 01
  0.1475000E 03
  0.4680000E 00
  0.9520000E-01
  0.1073261E 03
  0.3614712E 00
  0.2797779E-03
 -0.1140415E-07
 -0.2335579E-10
  0.5429524E-14
  0.8409600E 04
  0.7786586E 01
N-OCTANE
  0.1200000E 02
  0.1142000E 03
  0.3621000E 03
  0.1024900E 04
  0.7790000E 01
```

```
  0.7182000E 03
  0.3992000E 00
  0.7551000E 01
  0.1635000E 03
  0.4760000E 00
  0.9680000E-01
  0.1062346E 03
  0.3603365E 00
  0.2808558E-03
 -0.1416608E-07
 -0.2197521E-10
  0.5185566E-14
  0.9221000E 04
  0.7894018E 01
N-NONANE
  0.1300000E 02
  0.1283000E 03
  0.3320000E 03
  0.1073000E 04
  0.8710000E 01
  0.7634000E 03
  0.4439000E 00
  0.7650000E 01
  0.1796000E 03
  0.0000000E 00
  0.0000000E 00
  0.1054379E 03
  0.3594536E 00
  0.2815377E-03
 -0.1614648E-07
 -0.2097894E-10
  0.5010155E-14
  0.1045690E 05
  0.8332532E 01
N-DECANE
  0.1400000E 02
  0.1423000E 03
  0.3040000E 03
  0.1114000E 04
  0.9660000E 01
  0.8054000E 03
  0.4869000E 00
  0.7720000E 01
  0.1960000E 03
  0.0000000E 00
  0.0000000E 00
  0.1047748E 03
  0.3585556E 00
  0.2822217E-03
 -0.1709048E-07
 -0.2088661E-10
  0.5052078E-14
  0.1091200E 05
  0.8248089E 01
N-UNDECANE
  0.1500000E 02
  0.1563000E 03
```

```
 0.2820000E 03
 0.1154000E 04
 0.1060000E 02
 0.8446000E 03
 0.5210000E 00
 0.7790000E 01
 0.2122000E 03
 0.0000000E 00
 0.0000000E 00
 0.1042761E 03
 0.3578910E 00
 0.2825389E-03
-0.1812472E-07
-0.2036251E-10
 0.4956156E-14
 0.1148170E 05
 0.8260497E 01
N-DODECANE
 0.1600000E 02
 0.1700000E 03
 0.2620000E 03
 0.1188000E 04
 0.1150000E 02
 0.8813000E 03
 0.5610000E 00
 0.7840000E 01
 0.2286000E 03
 0.0000000E 00
 0.0000000E 00
 0.1037498E 03
 0.3577219E 00
 0.2831114E-03
-0.2057550E-07
-0.1859585E-10
 0.4555136E-14
 0.1185770E 05
 0.8150997E 01
N-TRIDECANE
 0.1700000E 02
 0.1840000E 03
 0.2500000E 03
 0.1220000E 04
 0.1200000E 02
 0.9158000E 03
 0.6002000E 00
 0.7890000E 01
 0.2449000E 03
 0.0000000E 00
 0.0000000E 00
 0.1033792E 03
 0.3570239E 00
 0.2835027E-03
-0.2048454E-07
-0.1919438E-10
 0.4754124E-14
 0.1299130E 05
 0.8491732E 01
```

```
N-TETRADECANE
  0.1800000E 02
  0.1984000E 03
  0.2350000E 03
  0.1251000E 04
  0.1300000E 02
  0.9483000E 03
  0.6399000E 00
  0.7920000E 01
  0.2613000E 03
  0.0000000E 00
  0.0000000E 00
  0.1030777E 03
  0.3566963E 00
  0.2836664E-03
 -0.2140798E-07
 -0.1857527E-10
  0.4619380E-14
  0.1375000E 05
  0.8628699E 01
N-PENTADECANE
  0.1900000E 02
  0.2124000E 03
  0.2200000E 03
  0.1278000E 04
  0.1400000E 02
  0.9791000E 03
  0.6743000E 00
  0.7960000E 01
  0.2778000E 03
  0.0000000E 00
  0.0000000E 00
  0.1027884E 03
  0.3562666E 00
  0.2840076E-03
 -0.2165476E-07
 -0.1874164E-10
  0.4695110E-14
  0.1463590E 05
  0.8822087E 01
N-HEXADECANE
  0.2000000E 02
  0.2264000E 03
  0.2060000E 03
  0.1303000E 04
  0.1500000E 02
  0.1008200E 04
  0.7078000E 00
  0.7990000E 01
  0.2941000E 03
  0.0000000E 00
  0.0000000E 00
  0.1025452E 03
  0.3560591E 00
  0.2842026E-03
 -0.2246747E-07
 -0.1822142E-10
```

0.4585024E-14
0.1540550E 05
0.8956267E 01
N-HEPTADECANE
0.2100000E 02
0.2405000E 03
0.1910000E 03
0.1328000E 04
0.1600000E 02
0.1034950E 04
0.7327000E 00
0.8030000E 01
0.3104000E 03
0.0000000E 00
0.0000000E 00
0.1023173E 03
0.3557580E 00
0.2844366E-03
-0.2282104E-07
-0.1816423E-10
0.4595705E-14
0.1560850E 05
0.8847407E 01
ETHYLENE
0.2200000E 02
0.2810000E 02
0.7421000E 03
0.5095100E 03
0.1980000E 01
0.3053000E 03
0.9490000E-01
0.6080000E 01
0.6100000E 02
0.4323000E 00
0.8760000E-01
0.1333033E 03
0.3579026E 00
0.1832974E-03
0.4920709E-07
-0.4794824E-10
0.9551773E-14
0.3453700E 04
0.7298306E 01
PROPYLENE
0.2300000E 02
0.4210000E 02
0.6670000E 03
0.6572000E 03
0.2900000E 01
0.4061000E 03
0.1451000E 00
0.6430000E 01
0.7900000E 02
0.4370000E 00
0.8890000E-01
0.1110291E 03
0.3374762E 00

0.2233077E-03
0.1396835E-07
-0.3049887E-10
0.6364897E-14
0.4697400E 04
0.7397910E 01
1-BUTENE
0.2400000E 02
0.5610000E 02
0.5830000E 03
0.7553000E 03
0.3870000E 01
0.4807000E 03
0.2085000E 00
0.6760000E 01
0.9530000E 02
0.4420000E 00
0.9020000E-01
0.1044740E 03
0.3380802E 00
0.2469572E-03
-0.2654500E-08
-0.2522633E-10
0.5734099E-14
0.5996700E 04
0.7826754E 01
CIS-2-BUTENE
0.2500000E 02
0.5610000E 02
0.6000000E 03
0.7708000E 03
0.2820000E 01
0.4987000E 03
0.2575000E 00
0.6760000E 01
0.9120000E 02
0.0000000E 00
0.0000000E 00
0.1011755E 03
0.3123419E 00
0.2274942E-03
0.3306823E-07
-0.4389652E-10
0.9073710E-14
0.6401000E 04
0.7976585E 01
TRANS-2-BUTENE
0.2600000E 02
0.5610000E 02
0.6000000E 03
0.7629000E 03
0.2820000E 01
0.4936000E 03
0.2230000E 00
0.6760000E 01
0.9380000E 02
0.0000000E 00

```
  0.0000000E 00
  0.1066749E 03
  0.3410696E 00
  0.2403610E-03
 -0.5046630E-08
 -0.2077516E-10
  0.4567499E-14
  0.6221600E 04
  0.7882542E 01
I-BUTENE
  0.2700000E 02
  0.5610000E 02
  0.5798000E 03
  0.7522000E 03
  0.3850000E 01
  0.4796000E 03
  0.1975000E 00
  0.6760000E 01
  0.9540000E 02
  0.0000000E 00
  0.0000000E 00
  0.1026202E 03
  0.3464635E 00
  0.2573637E-03
 -0.2924334E-07
 -0.8367410E-11
  0.2354519E-14
  0.5742900E 04
  0.7601563E 01
1,3-BUTADIENE
  0.2800000E 02
  0.5410000E 02
  0.6280000E 03
  0.7650000E 03
  0.3540000E 01
  0.4841000E 03
  0.2028000E 00
  0.6940000E 01
  0.8800000E 02
  0.0000000E 00
  0.0000000E 00
  0.9397795E 02
  0.3269950E 00
  0.2505054E-03
 -0.4180178E-07
 -0.7633850E-11
  0.3136749E-14
  0.5688200E 04
  0.7437603E 01
1-PENTENE
  0.2900000E 02
  0.7010000E 02
  0.5860000E 03
  0.8530000E 03
  0.4970000E 01
  0.5459000E 03
  0.2198000E 00
```

0.7050000E 01
0.1104000E 03
0.0000000E 00
0.0000000E 00
0.1065214E 03
0.3453769E 00
0.2505594E-03
-0.6887860E-08
-0.2267384E-10
0.5204260E-14
0.6931200E 04
0.7914969E 01
CIS-2-PENTENE
0.3000000E 02
0.7010000E 02
0.5188000E 03
0.8660000E 03
0.4850000E 01
0.5585000E 03
0.2060000E 00
0.7050000E 01
0.1078000E 03
0.0000000E 00
0.0000000E 00
0.9800248E 02
0.3214593E 00
0.2498968E-03
0.1079278E-07
-0.3455767E-10
0.7580039E-14
0.6876620E 04
0.7727490E 01
TRANS-2-PENTENE
0.3100000E 02
0.7010000E 02
0.5158000E 03
0.8630000E 03
0.4740000E 01
0.5574000E 03
0.2090000E 00
0.7050000E 01
0.1090000E 03
0.0000000E 00
0.0000000E 00
0.1017833E 03
0.3374028E 00
0.2530716E-03
-0.1027386E-07
-0.1993506E-10
0.4490075E-14
0.6841200E 04
0.7711650E 01
2METHYL-1BUTENE
0.3200000E 02
0.7010000E 02
0.5144000E 03
0.8500000E 03

```
  0.4760000E 01
  0.5481000E 03
  0.2000000E 00
  0.7050000E 01
  0.1087000E 03
  0.0000000E 00
  0.0000000E 00
  0.9988385E 02
  0.3473688E 00
  0.2711635E-03
 -0.3300913E-07
 -0.1040896E-10
  0.3190076E-14
  0.6474600E 04
  0.7751419E 01
3METHYL-1BUTENE
  0.3300000E 02
  0.7010000E 02
  0.5070000E 03
  0.8310000E 03
  0.5150000E 01
  0.5281000E 03
  0.1490000E 00
  0.7050000E 01
  0.1128000E 03
  0.0000000E 00
  0.0000000E 00
  0.1018051E 03
  0.3673974E 00
  0.2850419E-03
 -0.6718007E-07
  0.9834730E-11
 -0.7078140E-15
  0.6304660E 04
  0.7580100E 01
2METHYL-2BUTENE
  0.3400000E 02
  0.7010000E 02
  0.5276000E 03
  0.8700000E 03
  0.4650000E 01
  0.5614000E 03
  0.2120000E 00
  0.7050000E 01
  0.1067000E 03
  0.0000000E 00
  0.0000000E 00
  0.9735060E 02
  0.3266569E 00
  0.2529080E-03
 -0.6824350E-08
 -0.2187545E-10
  0.4904797E-14
  0.7112800E 04
  0.7889718E 01
1-HEXENE
  0.3500000E 02
```

```
  0.8420000E 02
  0.4712000E 03
  0.9200000E 03
  0.5890000E 01
  0.6063000E 03
  0.2463000E 00
  0.7400000E 01
  0.1258000E 03
  0.0000000E 00
  0.0000000E 00
  0.1050546E 03
  0.3461879E 00
  0.2567267E-03
 -0.1116226E-07
 -0.2072409E-10
  0.4803384E-14
  0.7787600E 04
  0.7930324E 01
CYCLOPENTANE
  0.3600000E 02
  0.7010000E 02
  0.6547000E 03
  0.9211700E 03
  0.4160000E 01
  0.5807000E 03
  0.2051000E 00
  0.8110000E 01
  0.9470000E 02
  0.0000000E 00
  0.0000000E 00
  0.7070346E 02
  0.2599424E 00
  0.2666641E-03
  0.4474819E-07
 -0.5736750E-10
  0.1188439E-13
  0.7411100E 04
  0.7940722E 01
METHYLCYCLOC5H12
  0.3700000E 02
  0.8410000E 02
  0.5490000E 03
  0.9589900E 03
  0.5110000E 01
  0.6213000E 03
  0.2346000E 00
  0.7850000E 01
  0.1131000E 03
  0.0000000E 00
  0.0000000E 00
  0.7850995E 02
  0.2858833E 00
  0.2735821E-03
  0.2352464E-07
 -0.4576194E-10
  0.9802556E-14
  0.7940000E 04
```

```
  0.7945471E 01
CYCLOHEXANE
  0.3800000E 02
  0.8420000E 02
  0.5610000E 03
  0.9953300E 03
  0.4980000E 01
  0.6373000E 03
  0.2032000E 00
  0.8200000E 01
  0.1087000E 03
  0.4440000E 00
  0.9030000E-01
  0.6802744E 02
  0.2696728E 00
  0.2956549E-03
  0.3563189E-07
 -0.5587736E-10
  0.1164154E-13
  0.7830900E 04
  0.7662126E 01
METHYLCYCLOC6H14
  0.3900000E 02
  0.9820000E 02
  0.5044000E 03
  0.1030200E 04
  0.5900000E 01
  0.6737000E 03
  0.2421000E 00
  0.7830000E 01
  0.1283000E 03
  0.0000000E 00
  0.0000000E 00
  0.7162929E 02
  0.2927432E 00
  0.3103910E-03
  0.3809860E-08
 -0.3989976E-10
  0.8989408E-14
  0.8549200E 04
  0.7909762E 01
BENZENE
  0.4000000E 02
  0.7810000E 02
  0.7140000E 03
  0.1012700E 04
  0.4180000E 01
  0.6362000E 03
  0.2130000E 00
  0.9160000E 01
  0.8940000E 02
  0.4450000E 00
  0.9040000E-01
  0.5927487E 02
  0.2298265E 00
  0.2230203E-03
 -0.7727730E-08
```

-0.2498888E-10
0.6114972E-14
0.8146500E 04
0.7833714E 01
TOLUENE
0.4100000E 02
0.9210000E 02
0.5900000E 03
0.1069200E 04
0.5200000E 01
0.6911000E 03
0.2591000E 00
0.8920000E 01
0.1068000E 03
0.0000000E 00
0.0000000E 00
0.6380814E 02
0.2447844E 00
0.2314995E-03
-0.1154979E-07
-0.2244468E-10
0.5561371E-14
0.8580500E 04
0.7719392E 01
O-XYLENE
0.4200000E 02
0.1062000E 03
0.5300000E 03
0.1138000E 04
0.6200000E 01
0.7519000E 03
0.2904000E 00
0.8990000E 01
0.1212000E 03
0.0000000E 00
0.0000000E 00
0.7224688E 02
0.2697178E 00
0.2368157E-03
-0.2556033E-07
-0.1292348E-10
0.3640964E-14
0.9998500E 04
0.8147551E 01
P-XYLENE
0.4300000E 02
0.1062000E 03
0.5000000E 03
0.1112800E 04
0.5900000E 01
0.7410000E 03
0.2969000E 00
0.8770000E 01
0.1240000E 03
0.0000000E 00
0.0000000E 00
0.6947142E 02

0.2579543E 00
0.2321720E-03
-0.1244061E-07
-0.2011161E-10
0.4897915E-14
0.9809900E 04
0.8124805E 01
M-XYLENE
0.4400000E 02
0.1062000E 03
0.5100000E 03
0.1114600E 04
0.6200000E 01
0.7424000E 03
0.3045000E 00
0.8820000E 01
0.1235000E 03
0.0000000E 00
0.0000000E 00
0.6880726E 02
0.2595038E 00
0.2361899E-03
-0.1685006E-07
-0.1802897E-10
0.4508186E-14
0.9904200E 04
0.8167049E 01
ETHYLBENZENE
0.4500000E 02
0.1062000E 03
0.5400000E 03
0.1115500E 04
0.5900000E 01
0.7371000E 03
0.2936000E 00
0.8790000E 01
0.1231000E 03
0.0000000E 00
0.0000000E 00
0.6861154E 02
0.2630397E 00
0.2436472E-03
-0.2044219E-07
-0.1886264E-10
0.5020942E-14
0.9301300E 04
0.7809470E 01

Property Data File

PPTEXTRA

```
    45
    30
HYDROGEN
  0.0000000E 00
  0.7000000E-01
 -0.4216000E 03
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.4383729E 00
 -0.2171200E-01
  0.3856489E-03
  0.8629312E 01
  0.2670000E 02
  0.9670000E-01
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.6670000E 02
  0.2959000E 01
  0.0000000E 00
  0.1000000E 01
  0.2700000E 01
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.3000000E 01
  0.0000000E 00
  0.3888000E 03
  0.3100000E 02
METHANE
  0.2700000E 03
  0.3421000E 00
  0.5200000E-03
  0.1080000E 02
 -0.5440000E 02
  0.1977000E-03
  0.2640000E 01
  0.7765430E-01
 -0.1121620E-02
 -0.1207200E-05
  0.7986220E 00
  0.1915903E 02
  0.4871330E 00
  0.4911648E-02
  0.2211270E-04
  0.3730119E-07
  0.0000000E 00
  0.2520000E 03
  0.3808000E 01
  0.2044096E-01
  0.5583634E-02
  0.5630233E-04
  0.2322720E-06
```

0.5104369E-09
0.0000000E 00
0.0000000E 00
0.1000000E 01
0.0000000E 00
0.3519000E 04
0.7260000E 02
ETHANE
0.4750000E 03
0.5166000E 00
0.5500000E-03
0.1080000E 02
0.1508000E 03
0.8675000E-04
0.3190000E 01
0.8300870E-01
-0.1891880E-02
0.1474410E-04
0.1437280E 00
0.8045569E 00
0.9207847E-02
0.1653079E-03
0.1271182E-05
0.3302101E-08
0.0000000E 00
0.4248000E 03
0.4384000E 01
0.6217944E-01
-0.1509117E-02
-0.1432870E-04
0.6807195E-06
0.2158669E-07
0.1893508E-09
0.5433102E-12
0.2000000E 01
0.0000000E 00
0.6330600E 04
0.1105000E 03
PROPANE
0.5950000E 03
0.5922000E 00
0.5400000E-03
0.1080000E 02
0.2642000E 03
0.1637000E-03
0.2910000E 01
0.8801200E-01
-0.2415530E-02
0.2551270E-04
-0.3362300E-01
0.6587896E 00
0.1654678E-02
0.1137806E-04
0.1437305E-06
0.8103741E-09
0.0000000E 00
0.3708000E 03

0.5240000E 01
0.1383535E 00
-0.1377435E-02
0.1463020E-04
-0.9389845E-07
-0.1926904E-08
0.3599645E-11
0.1363475E-12
0.2000000E 01
0.0000000E 00
0.8076600E 04
0.1508000E 03
I-BUTANE
0.6600000E 03
0.6339000E 00
0.5400000E-03
0.1080000E 02
0.3362000E 03
0.2390000E-03
0.2700000E 01
0.9043280E-01
-0.2681230E-02
0.3108580E-04
-0.7246600E-01
0.5521632E 00
0.1064005E-02
0.3342075E-05
-0.9518259E-08
0.7284382E-09
0.0000000E 00
0.3906000E 03
0.5819000E 01
0.2104895E 00
-0.2251166E-02
0.1579254E-04
-0.3613054E-06
0.3496504E-08
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.1000000E 01
0.9160200E 04
0.1910000E 03
N-BUTANE
0.6900000E 03
0.6531000E 00
0.4800000E-03
0.1260000E 02
0.3668000E 03
0.2973000E-03
0.2530000E 01
0.8732840E-01
-0.2446140E-02
0.2632970E-04
0.1660000E-03
0.5731460E 00
0.1086977E-02

```
  0.2509684E-05
  0.1996778E-07
  0.3331534E-09
  0.0000000E 00
  0.3744000E 03
  0.5869000E 01
  0.2050021E 00
 -0.1604553E-02
  0.1696180E-04
 -0.1480565E-06
  0.5101389E-09
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.0000000E 00
  0.9633600E 04
  0.1903000E 03
I-PENTANE
  0.7600000E 03
  0.6822000E 00
  0.4900000E-03
  0.1080000E 02
  0.4316000E 03
  0.1780000E-03
  0.2900000E 01
  0.8868990E-01
 -0.2557060E-02
  0.2848660E-04
 -0.6579600E-01
  0.5211352E 00
  0.8671523E-03
  0.3610140E-05
 -0.5147630E-07
  0.4953380E-09
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.2792657E 00
 -0.2872296E-02
  0.2095280E-04
 -0.4411422E-06
  0.4924242E-08
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.1000000E 01
  0.1062180E 05
  0.2300000E 03
N-PENTANE
  0.7700000E 03
  0.6917000E 00
  0.4800000E-03
  0.1260000E 02
  0.4478000E 03
  0.1972000E-03
  0.2760000E 01
  0.8726140E-01
```

```
-0.2492770E-02
 0.2741940E-04
-0.4821000E-02
 0.5383160E 00
 0.1380754E-02
 0.2462121E-06
 0.8585859E-07
-0.1893939E-09
 0.0000000E 00
 0.4842000E 03
 0.6099000E 01
 0.2765035E 00
-0.2634479E-02
 0.2255245E-04
-0.1361208E-06
 0.2913753E-09
 0.0000000E 00
 0.0000000E 00
 0.2000000E 01
 0.0000000E 00
 0.1108800E 05
 0.2310000E 03
NEO-PENTANE
 0.7100000E 03
 0.6649000E 00
 0.4900000E-03
 0.1260000E 02
 0.3830000E 03
 0.2830000E-03
 0.2500000E 01
 0.9447680E-01
-0.2983870E-02
 0.3655240E-04
-0.1215230E 00
 0.5383160E 00
 0.1380754E-02
 0.2462121E-06
 0.8585859E-07
-0.1893939E-09
 0.0000000E 00
 0.0000000E 00
 0.0000000E 00
 0.0000000E 00
 0.1000000E 01
 0.2834000E 03
 0.0000000E 00
 0.0000000E 00
 0.0000000E 00
 0.0000000E 00
 0.2000000E 01
 0.2000000E 01
 0.9788400E 04
 0.2310000E 03
N-HEXANE
 0.8400000E 03
 0.7198000E 00
 0.4600000E-03
```

0.1260000E 02
0.5162000E 03
0.2625000E-03
0.2590000E 01
0.8765670E-01
-0.2563440E-02
0.2904250E-04
-0.1970000E-01
0.5207786E 00
0.8422300E-03
0.1901224E-05
0.8207071E-08
0.3642193E-11
0.0000000E 00
0.7614000E 03
0.5916000E 01
0.3889860E 00
-0.3983654E-02
0.2336320E-04
-0.6767191E-07
0.4188520E-10
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.0000000E 00
0.1241280E 05
0.2710000E 03
N-HEPTANE
0.9000000E 03
0.7409000E 00
0.4200000E-03
0.1440000E 02
0.5738000E 03
0.2500000E-03
0.2600000E 01
0.8763000E-01
-0.2586870E-02
0.2959080E-04
-0.2165300E-01
0.5119534E 00
0.6884324E-03
0.1306090E-05
0.1099942E-07
-0.2367424E-10
0.0000000E 00
0.3204000E 03
0.7621000E 01
0.5301949E 00
-0.7683619E-02
0.8074276E-04
-0.4631362E-06
0.9964478E-09
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.0000000E 00
0.1363500E 05

```
  0.3110000E 03
N-OCTANE
 -0.9700000E 03
  0.7577000E 00
  0.4200000E-03
  0.1440000E 02
  0.6260000E 03
  0.0000000E 00
  0.0000000E 00
  0.8770230E-01
 -0.2614540E-02
  0.3024930E-04
 -0.2414300E-01
  0.5076503E 00
  0.7195998E-03
  0.1504953E-05
 -0.4564879E-08
  0.3277972E-10
  0.0000000E 00
  0.5994000E 03
  0.7407000E 01
  0.7041696E 00
 -0.9687067E-02
  0.8484951E-04
 -0.4204724E-06
  0.8507076E-09
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.0000000E 00
  0.1480320E 05
  0.3510000E 03
N-NONANE
 -0.1020000E 04
  0.7697000E 00
  0.4100000E-03
  0.1440000E 02
  0.6746000E 03
  0.0000000E 00
  0.0000000E 00
  0.8755860E-01
 -0.2616880E-02
  0.3032240E-04
 -0.2245900E-01
  0.5107070E 00
  0.6797110E-03
  0.3760190E-05
 -0.2294210E-07
  0.3620810E-09
 -0.7441410E-11
  0.4788000E 03
  0.8302000E 01
  0.0000000E 00
  0.2000000E 01
  0.8198000E 00
  0.0000000E 00
  0.0000000E 00
```

```
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.0000000E 00
  0.1587960E 05
  0.3910000E 03
N-DECANE
  0.1040000E 04
  0.7868000E 00
  0.4200000E-03
  0.1440000E 02
  0.7196000E 03
  0.2357000E-03
  0.2540000E 01
  0.8756570E-01
 -0.2630090E-02
  0.3062910E-04
 -0.2362400E-01
  0.5105150E 00
  0.5974280E-03
  0.5273100E-05
 -0.2731620E-07
 -0.4182030E-09
  0.5162670E-11
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.8724000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.0000000E 00
  0.1689660E 05
  0.4310000E 03
N-UNDECANE
  0.0000000E 00
  0.7401700E 00
  0.6800000E 02
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.8745910E-01
 -0.2629860E-02
  0.3060000E-04
 -0.2240400E-01
  0.5115150E 00
  0.4851940E-03
  0.6471910E-05
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
```

```
  0.0000000E 00
  0.2000000E 01
  0.9003000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.0000000E 00
  0.1785600E 05
  0.4708000E 03
N-DODECANE
  0.1115000E 04
  0.8021000E 00
  0.3800000E-03
  0.1800000E 02
  0.7988000E 03
  0.3260000E-03
  0.2200000E 01
  0.8754220E-01
 -0.2646670E-02
  0.3101410E-04
 -0.2453400E-01
  0.5131460E 00
  0.4678040E-03
  0.4257370E-05
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.9506000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.0000000E 00
  0.1877400E 05
  0.5105000E 03
N-TRIDECANE
  0.0000000E 00
  0.7564000E 00
  0.6800000E 02
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.8758680E-01
 -0.2659440E-02
  0.3133480E-04
 -0.2565500E-01
  0.5148630E 00
  0.4165320E-03
  0.4288650E-05
```

0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.9660000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.0000000E 00
0.1963800E 05
0.5506000E 03
N-TETRADECANE
0.0000000E 00
0.7628000E 00
0.6800000E 02
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.8740700E-01
-0.2647200E-02
0.3102170E-04
-0.2307800E-01
0.5147720E 00
0.5398050E-03
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.1002000E 01
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.0000000E 00
0.2048400E 05
0.5914000E 03
N-PENTADECANE
0.0000000E 00
0.7685000E 00
0.6800000E 02
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.8751550E-01
-0.2664740E-02

0.3147360E-04
-0.2512500E-01
0.5199950E 00
0.1872840E-03
0.6635460E-05
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.1038000E 01
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.0000000E 00
0.2127600E 05
0.6315000E 03
N-HEXADECANE
0.0000000E 00
0.7731000E 00
0.6800000E 02
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.8753000E-01
-0.2671040E-02
0.3163340E-04
-0.2558100E-01
0.5217420E 00
0.1363100E-03
0.7013500E-05
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.1074000E 01
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.0000000E 00
0.2203200E 05
0.6712000E 03
N-HEPTADECANE
0.0000000E 00
0.7780000E 00
0.6800000E 02
0.0000000E 00

0.0000000E 00
0.0000000E 00
0.0000000E 00
0.8753980E-01
-0.2676960E-02
0.3178830E-04
-0.2603600E-01
0.5230000E 00
0.8500000E-04
0.7380000E-05
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.1110000E 01
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.0000000E 00
0.2275200E 05
0.7110000E 03
ETHYLENE
0.4400000E 03
0.5222000E 00
0.6100000E-03
0.1260000E 02
0.1112000E 03
0.2130000E-03
0.2730000E 01
0.7738070E-01
-0.2337480E-02
0.2804980E-04
0.1013940E 00
0.1000417E 01
0.9018173E-02
0.9192708E-04
0.4869002E-08
0.1006155E-08
0.0000000E 00
0.4140000E 03
0.4066000E 01
0.7611536E-01
0.2058882E-02
0.8486484E-04
0.9433242E-06
0.3804823E-08
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.0000000E 00
0.5826600E 04
0.1000000E 03

PROPYLENE
-0.6000000E 03
0.6193000E 00
0.5700000E-03
0.1260000E 02
0.2588000E 03
0.0000000E 00
0.0000000E 00
0.7614230E-01
-0.2128240E-02
0.2300490E-04
0.6605800E-01
0.6320767E 00
0.1598205E-02
0.1433381E-04
0.1626314E-08
0.8092642E-09
0.0000000E 00
0.5454000E 03
0.4670000E 01
0.1035680E 00
-0.4479878E-03
0.6078914E-05
-0.1537747E-06
-0.7045263E-09
0.1382909E-10
0.1124907E-12
0.2000000E 01
0.0000000E 00
0.7923600E 04
0.1400000E 03
1-BUTENE
0.6850000E 03
0.6746000E 00
0.5400000E-03
0.1260000E 02
0.3560000E 03
0.3221000E-03
0.2560000E 01
0.8102150E-01
-0.2441940E-02
0.2890490E-04
-0.2134200E-01
0.5583959E 00
0.1558968E-02
0.3564994E-05
-0.2586339E-08
0.2453692E-08
0.0000000E 00
0.5742000E 03
0.5198000E 01
0.1824668E 00
-0.1629006E-02
0.8216296E-05
-0.5431920E-07
0.1185072E-08
-0.6558343E-11

0.0000000E 00
0.2000000E 01
0.0000000E 00
0.9428400E 04
0.1800000E 03
CIS-2-BUTENE
-0.7300000E 03
0.6983000E 00
0.5000000E-03
0.1260000E 02
0.3722000E 03
0.0000000E 00
0.0000000E 00
0.7553070E-01
-0.1998340E-02
0.1966530E-04
-0.7493000E-01
0.5432776E 00
0.1303716E-02
0.5810011E-05
-0.1103649E-08
0.9929379E-09
0.0000000E 00
0.4662000E 03
0.5508000E 01
0.2172436E 00
-0.1930090E-02
0.2124387E-04
-0.1749580E-06
0.5095893E-09
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.0000000E 00
0.1004400E 05
0.1777000E 03
TRANS-2-BUTENE
-0.7200000E 03
0.6800000E 00
0.5100000E-03
0.1260000E 02
0.3758000E 03
0.0000000E 00
0.0000000E 00
0.7650210E-01
-0.2109560E-02
0.2234660E-04
0.1128280E 00
0.5286554E 00
0.1419120E-02
0.8059479E-05
-0.2192708E-08
0.1553395E-08
0.0000000E 00
0.4662000E 03
0.5508000E 01
0.2172436E 00

-0.1930090E-02
0.2124387E-04
-0.1749580E-06
0.5095893E-09
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.0000000E 00
0.9790200E 04
0.1777000E 03
I-BUTENE
0.6800000E 03
0.6747000E 00
0.5100000E-03
0.1260000E 02
0.3542000E 03
0.2078000E-03
0.2880000E 01
0.7981960E-01
-0.2369230E-02
0.2767180E-04
0.8079000E-01
0.5294406E 00
0.9646853E-03
0.3275058E-05
0.4895105E-07
-0.2331002E-09
0.0000000E 00
0.7650000E 03
0.4776000E 01
0.1677786E 00
-0.7117552E-03
0.9589551E-05
-0.2061716E-06
0.1501764E-08
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.1000000E 01
0.9514800E 04
0.1791000E 03
1,3-BUTADIENE
0.6950000E 03
0.7014000E 00
0.5600000E-03
0.1260000E 02
0.3668000E 03
0.3081000E-03
0.2600000E 01
0.8340480E-01
-0.3117450E-02
0.4502670E-04
-0.7128700E-01
0.5436124E 00
0.8651076E-03
0.3801376E-06
0.3625266E-07

0.3171384E-09
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2011613E 00
-0.1845614E-02
0.8309090E-05
0.8656642E-07
0.4800000E-09
-0.1736213E-10
0.8614608E-13
0.2000000E 01
0.0000000E 00
0.9660600E 04
0.1982000E 03
1-PENTENE
0.0000000E 00
0.6405000E 00
0.6800000E 02
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.8181110E-01
-0.2457400E-02
0.2877700E-04
0.4115000E-02
0.5068205E 00
0.8912393E-03
-0.3933566E-07
-0.4220086E-07
0.3751457E-09
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2359794E 00
-0.1848246E-02
0.1755208E-04
-0.1535669E-06
0.6273674E-09
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.0000000E 00
0.1083960E 05
0.2182000E 03
CIS-2-PENTENE
0.0000000E 00
0.6556000E 00
0.6800000E 02
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00

```
  0.0000000E 00
  0.4947427E 00
  0.8389447E-03
  0.2601574E-05
 -0.1911173E-07
 -0.1086851E-09
 -0.2575322E-12
  0.3672000E 03
  0.6476000E 01
  0.0000000E 00
  0.2000000E 01
  0.6580000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.0000000E 00
  0.1123200E 05
  0.2177000E 03
TRANS-2-PENTENE
 -0.8100000E 03
  0.7149000E 00
  0.5000000E-03
  0.1260000E 02
  0.4568000E 03
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.5114111E 00
  0.8906539E-03
  0.2181304E-05
 -0.5450215E-08
 -0.3136979E-10
 -0.3960201E-13
  0.3672000E 03
  0.6476000E 01
  0.0000000E 00
  0.2000000E 01
  0.6580000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.0000000E 00
  0.1121400E 05
  0.2177000E 03
2METHYL-1BUTENE
  0.0000000E 00
  0.6504000E 00
  0.6800000E 02
  0.0000000E 00
  0.0000000E 00
```

0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.5131009E 00
0.8544243E-03
0.2308273E-05
-0.8074766E-08
-0.1856173E-10
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.6580000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.6094000E 01
3METHYL-1BUTENE
0.0000000E 00
0.6272000E 00
0.6800000E 02
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.5059818E 00
0.9587270E-03
0.2596944E-05
0.7525459E-08
0.6540221E-10
0.1669162E-12
0.5094000E 03
0.5829000E 01
0.0000000E 00
0.2000000E 01
0.6580000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.2000000E 01
0.0000000E 00
0.1035000E 05
0.2191000E 03
2METHYL-2BUTENE

```
  0.0000000E 00
  0.6623000E 00
  0.6800000E 02
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.4996087E 00
  0.7620027E-03
  0.2300449E-05
 -0.4041000E-08
  0.3777012E-10
  0.2326183E-13
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.6580000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.2000000E 01
  0.0000000E 00
  0.1131660E 05
  0.2169000E 03
1-HEXENE
  0.0000000E 00
  0.6731000E 00
  0.6800000E 02
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.8278840E-01
 -0.2506430E-02
  0.2956270E-04
 -0.3709000E-02
  0.4981515E 00
  0.8133644E-03
  0.1627331E-05
 -0.2831197E-07
  0.1784674E-09
  0.0000000E 00
  0.0000000E 00
  0.0000000E 00
  0.3579420E 00
 -0.4676289E-02
  0.5405289E-04
 -0.3956416E-06
  0.1118046E-08
  0.0000000E 00
  0.0000000E 00
```

0.2000000E 01
0.0000000E 00
0.1216800E 05
0.2591000E 03
CYCLOPENTANE
0.8500000E 03
0.8140000E 00
0.5100000E-03
0.1440000E 02
0.5234000E 03
0.2750000E-03
0.2700000E 01
0.7947290E-01
-0.2038850E-02
0.1853690E-04
-0.4635000E 00
0.4074428E 00
0.9357009E-03
0.1412078E-05
0.3734532E-08
-0.7155799E-10
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.5667362E 00
-0.6854083E-02
0.4830217E-04
-0.2005489E-06
0.3688378E-09
0.0000000E 00
0.0000000E 00
0.1000000E 01
0.0000000E 00
0.1174320E 05
0.2050000E 03
METHYLCYCLOC5H12
0.8850000E 03
0.8089000E 00
0.4800000E-03
0.1440000E 02
0.5612000E 03
0.2630000E-03
0.2700000E 01
0.8235980E-01
-0.2283010E-02
0.2366960E-04
-0.3561710E 00
0.4259522E 00
0.9408127E-03
0.2287533E-05
-0.1169671E-07
-0.6088833E-10
-0.1276287E-12
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.1000000E 01

```
 0.3036000E 03
 0.0000000E 00
 0.0000000E 00
 0.0000000E 00
 0.0000000E 00
 0.1000000E 01
 0.1000000E 01
 0.1260360E 05
 0.2428000E 03
CYCLOHEXANE
-0.9400000E 03
 0.8370000E 00
 0.4800000E-03
 0.1440000E 02
 0.5990000E 03
 0.0000000E 00
 0.0000000E 00
 0.8342310E-01
-0.2069790E-02
 0.1630780E-04
-0.4681440E 00
 0.4108182E 00
 0.1376486E-02
-0.1343677E-04
 0.1914335E-08
-0.7721443E-09
 0.0000000E 00
 0.5634000E 03
 0.6143000E 01
 0.1443039E 01
-0.2648453E-01
 0.2419787E-03
-0.9894205E-06
 0.1118439E-08
 0.0000000E 00
 0.0000000E 00
 0.1000000E 01
 0.0000000E 00
 0.1294200E 05
 0.2410000E 03
METHYLCYCLOC6H14
-0.9750000E 03
 0.8248000E 00
 0.4200000E-03
 0.1620000E 02
 0.6314000E 03
 0.0000000E 00
 0.0000000E 00
 0.8694620E-01
-0.2393080E-02
 0.2331080E-04
-0.3817010E 00
 0.4259582E 00
 0.9017870E-03
-0.1784924E-05
-0.8590629E-07
-0.6896876E-09
```

```
-0.1795573E-11
 0.0000000E 00
 0.0000000E 00
 0.0000000E 00
 0.1000000E 01
 0.3592000E 03
 0.0000000E 00
 0.0000000E 00
 0.0000000E 00
 0.0000000E 00
 0.1000000E 01
 0.1000000E 01
 0.1364400E 05
 0.2816000E 03
BENZENE
 0.9400000E 03
 0.9485000E 00
 0.5300000E-03
 0.1800000E 02
 0.6206000E 03
 0.4790000E-03
 0.2500000E 01
 0.6823610E-01
-0.2274850E-02
 0.2874360E-04
-0.2890640E 00
 0.4000699E 00
 0.8470820E-03
-0.3208770E-05
 0.2173173E-07
-0.3460080E-10
 0.0000000E 00
 0.6030000E 03
 0.5628000E 01
 0.9095834E 00
-0.1519808E-01
 0.1420271E-03
-0.6852134E-06
 0.1298083E-08
 0.0000000E 00
 0.0000000E 00
 0.1000000E 01
 0.0000000E 00
 0.1323540E 05
 0.2060000E 03
TOLUENE
 0.9950000E 03
 0.9314000E 00
 0.5000000E-03
 0.1800000E 02
 0.6782000E 03
 0.3352000E-03
 0.2680000E 01
 0.6918020E-01
-0.2191720E-02
 0.2648560E-04
-0.2239140E 00
```

0.4063217E 00
0.3641997E-03
0.1812355E-05
0.3836438E-08
-0.2913752E-10
0.0000000E 00
0.6786000E 03
0.5932000E 01
0.7616483E 00
-0.1050007E-01
0.9043425E-04
-0.4200967E-06
0.7664719E-09
0.0000000E 00
0.0000000E 00
0.1000000E 01
0.1000000E 01
0.1427940E 05
0.2460000E 03
O-XYLENE
0.1060000E 04
0.9430000E 00
0.4700000E-03
0.1800000E 02
0.7466000E 03
0.2986000E-03
0.2770000E 01
0.6949090E-01
-0.2136140E-02
0.2506570E-04
-0.7825800E-01
0.4140000E 00
0.4702700E-03
-0.1062791E-05
0.9639662E-08
-0.9105469E-11
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.1109943E 01
-0.1544222E-01
0.1130204E-03
-0.4120885E-06
0.5832065E-09
0.0000000E 00
0.0000000E 00
0.1000000E 01
0.2000000E 01
0.1584180E 05
0.2840000E 03
P-XYLENE
0.1025000E 04
0.9252000E 00
0.4800000E-03
0.1800000E 02
0.7286000E 03
0.1143000E-03

0.3350000E 01
0.6841050E-01
-0.2034540E-02
0.2278180E-04
-0.1359550E 00
0.3901189E 00
0.7576340E-03
-0.1124708E-05
0.1806526E-07
-0.5099066E-10
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.7943793E 00
-0.1043875E-01
0.8427413E-04
-0.3568050E-06
0.5809179E-09
0.0000000E 00
0.0000000E 00
0.1000000E 01
0.2000000E 01
0.1551600E 05
0.2840000E 03
M-XYLENE
0.1045000E 04
0.9275000E 00
0.4800000E-03
0.1800000E 02
0.7304000E 03
0.1844000E-03
0.3200000E 01
0.6962090E-01
-0.2126540E-02
0.2462690E-04
-0.1501940E 00
0.4033497E 00
0.2871989E-03
0.5244769E-07
0.2020202E-07
-0.6555943E-10
0.0000000E 00
0.0000000E 00
0.0000000E 00
0.7943793E 00
-0.1043875E-01
0.8427416E-04
-0.3568050E-06
0.5809179E-09
0.0000000E 00
0.0000000E 00
0.1000000E 01
0.2000000E 01
0.1566000E 05
0.2840000E 03
ETHYLBENZENE
0.1055000E 04

```
 0.9242000E 00
 0.4600000E-03
 0.1620000E 02
 0.7160000E 03
 0.1143000E-03
 0.3350000E 01
 0.7299140E-01
-0.2379330E-02
 0.2965190E-04
-0.2005730E 00
 0.4005524E 00
 0.3662684E-03
 0.1529740E-07
 0.9736790E-08
-0.2367424E-10
 0.0000000E 00
 0.0000000E 00
 0.0000000E 00
 0.8881474E 00
-0.1144597E-01
 0.8347160E-04
-0.3339025E-06
 0.5606698E-09
 0.0000000E 00
 0.0000000E 00
 0.1000000E 01
 0.2000000E 01
 0.1548000E 05
 0.2840000E 03
```