de Roulhac, Selma Lee

COMPARISON OF CHEMICAL PROCESS SIMULATION PROGRAMS FOR EDUCATION

The University of Arizona M.S. 1985

COMPARISON OF CHEMICAL PROCESS SIMULATION PROGRAMS FOR EDUCATION

by

Selma Lee de Roulhac

A Thesis Submitted to the Faculty of the

DEPARTMENT OF CHEMICAL ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 8 5

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department of the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Helena Teresa de Rentlow

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

Thomas R Rehm

Thomas R. Rehm
Professor of Chemical Engineering

6 May 85

Date

ACKNOWLEDGMENTS

Appreciation is extended to Dr. Thomas R. Rehm for his support and advice during this research.

A special thanks is also extended to Darde for his help.

TABLE OF CONTENTS

TABLE OF CONTENTS--Continued

LIST OF ILLUSTRATIONS

LIST OF TABLES

ABSTRACT

The purpose of this research was to evaluate three process simulation programs: CHESS, PROCESS, and FLOWTRAN for use in Chemical Engineering education. The basis for the evaluation was the program's abilities, software, student interface, and structure. The programs were also compared on their ability to simulate a simple dealkylation process.

The PROCESS program is very impressive, but it is too large for the University of Arizona's limited resources. The CHESS program, on the other hand, is smaller and easier to operate, but is too limited in it's abilities. The program recommended was the FLOWTRAN system. It contains a good balance between abilities and ease of use.

CHAPTER 1

INTRODUCTION

Simulation has been described by R. L. Motard, M. Shacham, and E. M. Rosen (1975) as "the representation of a chemical process by a mathematical model which was then solved to obtain information about the performance of the chemical process". The term flowsheeting system is a more specific term which L. B. Evans (1981) used to describe a system which "accepts, as input, information about a chemical process at the flowsheet level of detail and performs analyses useful in process development, design, or operation." A flowsheeting system could be specific to a particular process, or general enough to be applicable to most any process. It also could represent a steady state condition or vary with time.

A. W. Westerberg, H. P. Hutchison, R. L. Motard, and P. Winter (1979) have divided the development of flowsheeting systems in industry into four time periods. In the years 1955 through 1959, the few programs which were available were very limited and usually could only simulate one certain type of process equipment. Then, in the years 1960 through 1964, several industrial simulators appeared. Most of these were modular to allow the older single unit

programs to be adapted as subroutines. The proving and testing of these programs came from 1965 to 1969 when large scale computing power became readily available. The programs were not easily accepted at first due to not only a resistance to change but also their tremendous cost. According to Evans (1981), a typical development effort could require 20 to 60 man years and cost several million dollars. Management was hesitant to expend this much money all at once so simulators had to be developed gradually and proven financially all along the way. Finally, around 1970, process flowsheeting programs were accepted as being important, and useful. This acceptance has still continued. S. I. Proctor (1983), for example, estimated that FLOWTRAN was used approximately 2000 times per month in 1981.

To enable educators to better prepare students to use these programs, several have been made available to universities. At the time of this research, no process flowsheeting programs were in use at the University of Arizona. The purpose of this research was to evaluate three simulation programs: CHESS, PROCESS, and FLOWTRAN to determine which one would be best suited to use. Other comparisons of some of these programs have appeared in literature but never in enough detail.

R. L. Motard, et. al. (1975) reviewed the various aspects of data processing, the different approaches to engineering models, and the numerical procedures used in

flowsheeting programs. Unfortunately, these characteristics were never related to any existing programs.

L. B. Evans (1980) wrote an article similar to that of R. L. Motard, et. al. (1975) describing the essential elements of flowsheeting programs, except he included references to ASPEN. ASPEN was developed at the Massachusetts Institute of Technology. Evans also attempted to forecast the future characteristics of flowsheeting systems.

A series of articles written by E. T. Briddell (1974) also described a general flowsheeting program. Unfortunately, in the last article, the program example used was GPSS.

J. R. Flower and B. D. Whitehead (1973), on the other hand, listed characteristics of several programs including CHESS and FLOWTRAN. For CHESS, the article stated "a facility exists to punch intermediate results on cards", "an optimisation package has been linked with CHESS", and "CHESS overcomes some of the limitations of PACER and provides a physical property package and a simple convergence routine." The version of CHESS brought to this university could not punch cards, though, and did not have the optimization package linked to it.

For FLOWTRAN, the comments "a precompiler is used to check input data before execution", the "physical property scheme has been widely discussed in the literature and is

considered by many to be the best available", and "FLOWTRAN also has a large unit subroutine library" were given.

Jeffery N. Peterson, Chau-Chyen Chen, and Lawrence B. Evans (1978) listed most of the programs currently in use, and for each program gave a very short paragraph highlighting the abilities of the program and an address to write to for more information.

CHESS was described as a complete system containing an "integrated thermophysical property package" which "handles ideal and non-ideal organic systems in single- or two-phase," and blocks for both short-cut and rigorous fractionators, absorbers, pumps, compressors, expanders, heat exchangers, control blocks, and reactors.

The FLOWTRAN system was described as consisting of four parts: FLOWTRAN, PROPTY, VLE, and INF. These are discussed in detail in Chapter 6. Peterson also mentioned that FLOWTRAN contained data for 180 chemical species in it's public data file.

A panel discussion featuring GEPDS, FLOWTRAN, PACER 245, and PDA also appeared in the literature. (See W. D. Seider, 1972) In the discussion, each panelist gave a short description, discussed the support and training which would be provided, how proprietary information could be protected, whether it was wise to market, and how vapor-liquid equilibrium was handled for the program. Unfortunately, the

major emphasis was corporate strategy and not the programs themselves.

Articles which have appeared in the literature which discuss only FLOWTRAN include R. E. Harris (1972), Alvin H. Larsen (1982), Stanley I. Proctor (1983), Robert C. Rorschach and Robert E. Harris (1970), E. M. Rosen and D. J. Kaufman (1981), and E. M. Rosen and A. C. Pauls (1977). These articles described either the program or how the program was used to solve simulation problems.

An article which discussed only the CHESS program was written by P. Friedman, and K. L. Pinder (1972). In the article, several optimization models were added to the CHESS program to improve the operation of a process.

Very little about the PROCESS system has appeared in the literature because the system is proprietary and was not introduced until 1979. Two articles which were found were both written by N. F. Brannock, V. S. Verneuil, and Y. L. Wang. One was written in 1982 and the other in 1979. No comparisons appeared in either article, just descriptions of the PROCESS program.

Articles have also appeared which discussed the use of computer simulation programs in education. Some which were found were R. L. Motard and D. M. Himmelblau (1979); C. J. King, A. S. Foss, E. A. Grens, S. Lynn, and D. F. Rudd (1973); and D. R. Woods, et. al. (1973). All of these articles discussed how programs could be used in education,

as well as, some problems which were associated with using simulation programs.

The ideal comparison which was searched for would be similar to an appraisal made between PACER, GEMCS, and CONCEPT by N. Peters and P. E. Barker (1974). In their comparison, a short description of flowsheeting was given, an example worked for each program, and a comparison made of how the various characteristics affected the results.

This report follows the same organization. First, a description of how the programs were evaluated is given including a description of the methods used in flowsheeting programs, second, an example is described which was used to test each program, and third a final evaluation is discussed for each program, and some recommendations are made.

CHAPTER 2

EVALUATION BASIS

The evaluation of each of the three programs, CHESS, FLOWTRAN and PROCESS is divided into four categories: abilities, software, interface, and structure. The abilities section describes what the program is able to simulate, software is the numerical methods the program uses, interface is how the students interact with the program, and structure is how the software works on the computer.

Abilities

The abilities of interest are the models used to represent the process equipment, the range of operating conditions, the size of the physical property database, and the correlations used to determine the physical properties. If a program can simulate a greater number of units and components, it is applicable to more situations. It may also require more computer memory to execute.

Some programs combine variations of similar units into one model. Then, an option is given to specify which exact unit is desired. For example, some programs combine both compressors and pumps into one model. For this reason,

the total number of models a simulation package contains is not an important part of an evaluation basis.

The appropriateness of the models is considered. A very rigorous model is not needed if only a preliminary material balance is to be done. On the other hand, sometimes rigorous models are the ones which are needed. Rigorous models usually require additional input since additional output is calculated. There is no disadvantage to having both types of unit module routines provided both types do not have to be loaded into memory at once.

Models should also be able to operate in either of two modes: simulation and design. The simulation mode occurs when all the process input variables, and the equipment design factors are provided. It is also called the 'performance' or 'rating' mode. The simulation mode of operation is the most common since it is easier to write. This is because the program will have the same input data every time.

The design mode is almost the opposite. The design mode occurs when the outputs are specified and the system inputs and/or design parameters calculated. For most programs, a design mode calculation is simply an optimization on various simulation modes. This makes these calculations very time consuming. Design mode specifications do not have to apply to just one model. Some

programs have the option of controlling several units or entire processes in order to meet design criteria.

The size of the thermodynamic database is not as crucial for educational purposes as it is to industry. This is because educational users can choose a process to fit the database, whereas, the process is fixed in industry. The ideal choice, in this circumstance, will be a database which is large enough without too much memory being occupied by seldomly used compounds.

In an article by J. R. Fair (1980), a table was given comparing various databases including CHESS and FLOWTRAN. The total number of compounds, the number of physical property correlations, the addition of new compounds, and the ability to perform flash calculations were included in the table. No discussion was made on this information, though.

The type of thermodynamic correlations available is directly related to the range of operating conditions. Most programs only allow one method to be used at a time, but others allow different techniques to be used for different units. This is important when one unit's results depends heavily on the accuracy of the correlations, and the rest of the units's do not. If the program is to be used for instructional use only, educators usually require less accurate thermodynamic and physical property correlations than industry (Motard, et. al., 1975).

The book by Reid, Prausnitz, and Sherwood (1977) and the article by Nathan A. Massey (1982) both contained sections discussing thermodynamic correlation methods and their application under different circumstances. The Massey article discussed only the methods used in simulation programs, and some common difficulties associated with using the different methods.

Software

The numerical methods a program uses is referred to as software. The average simulation program follows five basic steps: 1. Read and check input data; 2. Determine the order of calculation; 3. Perform unit module calculations; 4. Check for recycle convergence; and 5. Output results. (Motard, et. al.,1975) Although steps 1 and 5 do involve software, they are discussed in the interface section.

Determining the order of calculation involves three operations. The first is partitioning. Partitioning was defined by R. L. Motard, et. al. (1975) as the identifying of recycle systems and ordering the nodes which are not included in those systems. Each partition will then be a unit or group of units which must be solved simultaneously.

The second operation is nesting or deciding how to identify recycle loops. Most programs default to ignoring nested loops. This prevents each of the inner loops from having to converge for each trial of the outer loop.

Tearing is the last operation. It involves choosing a stream on which to make an initial estimate, and the stream on which to apply the convergence criteria. There are many theories pertaining to which stream is the optimal one to tear. Most of these were discussed by A. W. Westerberg, et. al. (1979).

The choice of which criteria to use for tearing is fixed in programs which contain automatic tearing algorithms. Some problems Westerberg believed which were created by fixing the tearing algorithm, were that no single method was optimal every time, and deciding which stream to tear could take longer than the rest of the program.

To perform the unit module calculations three different approaches are possible. These are the sequential modular approach, the equation-oriented approach, and the simultaneous modular or 'two teir' approach.

In the sequential modular approach, each unit is represented by a separate subroutine, and the subroutines are called sequentially in a predetermined order. If a design specification is made involving more than one unit, it is solved iteratively. The same is true for recycle solutions. The sequential modular method is the most popular method because it allows the older single unit simulation programs to be adapted as subroutines. Sequential modular programs are also easier to write and

test because the work can be divided. One disadvantage to this approach is that it usually takes longer to run.

The equation-oriented approach involves describing the flowsheet with a series of non-linear algebraic equations. Much of the work in simulation programs today is involved in improving this approach and determining new procedures for solving the resultant sparse matrices. (See M. J. Mattione, W. J. Meier, and N. L. Book, 1982; M. Shacham, et. al., 1982; and B. A. Murtagh, 1982) L. B. Evans (1981) stated the advantages of using this approach were faster convergence, and easier optimization. The disadvantages were that it needed good starting values, a large investment had already been made in industry in unit operations models, and it was difficult to diagnose errors when the solution did not converge. Equation-oriented methods are used extensively in individual unit operations models but they have yet to be used routinely in any industrial flowsheeting programs.

In a comparison between sequential and equation-oriented approaches, T. Umeda and M. Nishio (1972) stated "it cannot be said that the convergence by the simultaneous approach is better than that by the sequential approach in any steady-state process simulation." The main problem which they saw in the simultaneous approach was assuring that the process was completely defined.

The 'two tier', or simultaneous modular algorithms, use two types of models, simple and rigorous (See Evans, 1980). The rigorous models are first used to determine the parameters for the simple linear models which are then solved simultaneously for all of the stream variables. These solutions are then used in the rigorous models. This process is repeated until a solution is reached. The simultaneous modular method overcomes the disadvantage the equation-oriented approach has of needing good starting values but not the other disadvantages. The computation time per iteration is likely to be larger than the modular approach, but the total number of iterations smaller (Motard, et. al., 1975). This approach does not require the order of the units in the loop to be known since the inputs to all of the rigorous units are known through the simple models.

The fourth step in a simulation program is checking for recycle convergence. Two methods used for accelerating the recycle convergence in simulation are Bounded Wegstein and Dominant Eigenvalue. When these methods are applied, they are applied only to the individual component flowrates.

Both Dominant Eigenvalue and Bounded Wegstein assume composition can be approximated by

$$X_{k+1} = qX_k + (1-q)\bar{X}_{k+1} \qquad (2.1)$$

where X_k is the estimated value at the beginning of trial k,

and $\bar{X}_{k+1}$ is the calculated value after trial k. Neither method can be applied until at least one trial involving direct substitution has been made. The difference between the two methods is how q, the acceleration parameter, is calculated.

The Bounded Wegstein method defines q as (See Motard, et. al., 1975)

$$q = w / (w-1) \tag{2.2}$$

where

$$w = (X_{k+1} - X_k) / (\bar{X}_k - \bar{X}_{k-1}) \tag{2.3}$$

This neglects the interaction between variables in the streams. The PROCESS manual (1983) stated that this technique was "not suitable for cases involving acceleration of more than one recycle stream and if applied may cause oscillation and hinder convergence." This is because the method is unable to distinguish between the effects of different accelerations. Despite this, according to Motard, et. al. (1975), this method is the most common and is used mainly because it is easy to code and quick.

The acceleration parameter, q, in the Dominant Eigenvalue method is defined as

$$q = \text{lambda} / (\text{lambda} - 1) \tag{2.4}$$

where

$$\text{lambda} = (\| X_{k+1} - X_k \|) / (\| X_k - X_{k-1} \|) \quad (2.5)$$

(See Motard, et. al., 1975; and O. Orbach and C. M. Crowe, 1971) The acceleration parameter now takes advantage of the interaction of the components in all of the recycle streams.

There are many other numerical methods involved in simulation programs. A book by L. M. Rose (1974) goes into more detail on the application of mathematical models to flowsheeting programs, and contains a good appendix of numerical methods and the suggested contents of a chemical engineering computer program library.

Interface

The input required, the output given, and the documentation provided are all parts of the interface. A program is of no use unless someone can learn how to use it. The requirements of a good interface, according to Evans (1980), were that it "accept the input in as natural a form as possible," and "present the results in as usable a form as possible."

There are several different types of format for input. Included in these are fixed format, free format with keywords, and FORTRAN NAMELIST. Examples of these are shown in Fig. 2.1.

Fixed Format

```
_1MIXRM201__1__2_-3__0__0__0
```

Free Format with Keywords

```
MIXER UID=1,NAME=M201
FEED  1,2
PRODUCT M=3
```

FORTRAN NAMELIST

```
_$KELIST
_KPM1=1,MIXR,M201,1,2,-3,3*0,
_$END
```

Figure 2.1 Input Data Examples

Fixed format data occurs when the data is required to appear in specific columns in a specific order. This type is better for large amounts of data, but is not good for programs which are not heavily used.

Free format data with keywords is easier to remember and learn, and therefore, would be much better for educational purposes. The only problem associated with this type of format is to make sure the keywords are easy to associate with the data. Free format data always requires a preprocessor, which increases the execution time, and the program size if the preprocessor is internal.

The FORTRAN NAMELIST function allows values for a variable to be inputted by giving the variable name = value. For arrays, the form is simply array name = value 1, value 2, etc. This type of data can be as good as free format or as bad as fixed format depending on the choice of variable names. Values for every variable do not have to be given if the values are set to zero prior to use. This saves time inputting the data. If an entire vector is to be inputted, the dimension of the array has to correspond to the number of variables given.

The quantity of output needed depends on the rigorousness of the simulation. Ideally, the normal output is short. Then, when needed, some variables are set which allow a more comprehensive output. There should also be some error checking done with messages printed when needed.

The cost of printing the output needs to be carefully compared with the value of having more information.

All programs come with some sort of documentation. This documentation should accurately reflect what the program does and needs. Having clear documentation is very important for educational uses since a new and different group of students would be using the program every year.

Structure

Normally, structure refers to the type of executive program a flowsheeting system contains. In this research, structure also includes how a program is loaded into memory when executing, how it is stored, and the speed of execution. The computer systems used are a CDC CYBER 175 with a NOS/BE operating system and a VAX 11/780 with a VMS operating system.

Westerberg, et. al. (1979) defined a flowsheeting executive as that part of the program which received the user input, collected the problem description, and then called executing, costing, and output routines. There are two types of executives, fixed and variable.

A fixed executive is one in which the executive program never changes. Therefore, the maximum size of process the program can handle will be fixed in the program. This increases the memory requirement of the program since the requirement will have to be set for the maximum size.

A variable structured program is one in which the main program is generated from the input data. This new main program is then compiled and linked with only the subroutines which are needed. The advantage of having a variable structured executive is that the arrays in the program will only be set to the amount needed for that particular process. This is better for simulating small plants because less memory is wasted on empty array elements.

The memory required for a program is also affected by how the program is loaded. Usually, an entire program is loaded into memory, and then executed. There are other ways. A library of subroutines could be kept, and a program could be segmented.

When a library of subroutines is kept, only the subroutines which are needed in the entire program are loaded. This usually reduces the memory requirement provided all of the subroutines are not needed.

If a program is segmented, only part of the program is loaded at one time. Which part is loaded depends on which part is needed at that point in the program. Segmenting a program uses less memory space in the computer than the usual loading procedure, but requires more computation time to allow for memory swapping according to Motard, et. al. (1975). Usually, segmentation is not done unless the program will not fit into memory, so this extra

time is of no concern. There are several ways of segmenting a program. The two most common are using overlays and SEGLOAD procedures.

When a program is segmented using overlays, the program is essentially divided into smaller programs, called overlays, which execute in a predetermined sequence. This sequence is contained in the main overlay, and the smaller programs are contained in the primary overlays. If a primary overlay calls several subroutines, these subroutines can be placed in secondary overlays. Computer memory is conserved using overlays because only one primary and one secondary overlay is in the computer memory at any given time.

SEGLOAD is a utility which can be used on the CYBER to control which portions of a program a written into the computer's memory at a given place in the program. These portions are defined by the programmer using directives and accessed similar to subroutines.

The ease of installation is very subjective. Preferably, a program should be able to be read from a tape and compiled to be ready for use. If a program is hard to install, the cost of installing the program is usually also very high.

Other costs which need to be considered are the cost of executing the program, and the cost of storing the program on the computer for student access. To store a

program for easy access, the program has to be placed in a permanent file. In 1984 on the University of Arizona CYBER, this cost is $0.32 per day for each thousand blocks of storage. In this circumstance it is better to have a small program.

CHAPTER 3

EXAMPLE PROCESS

To further illustrate the differences between the three programs, an example process is included in the comparison. Specifically, this example is meant to demonstrate the use,the versatility, and the speed and cost of execution of the programs.

The main advantage flowsheeting programs have over single unit or process-specific simulation programs is versatility. This versatility also causes several problems since some difficult situations the program could encounter cannot be compensated for in advance. Michael Leesley (1982) listed some of these situations as: material recycle loops; energy recycle loops; a wide range of operating temperatures, pressures, and components; a wide variety of process units, cost estimation models, and profitability analysis models; and the addition of compounds and unit operations. As many of these situations as possible are incorporated into the example process.

The process used is the dealkylation of toluene to form benzene. A process flowsheet is shown in Fig. 3.1. This example is a simplification of a process shown in Kirk-Othmer, 1964.

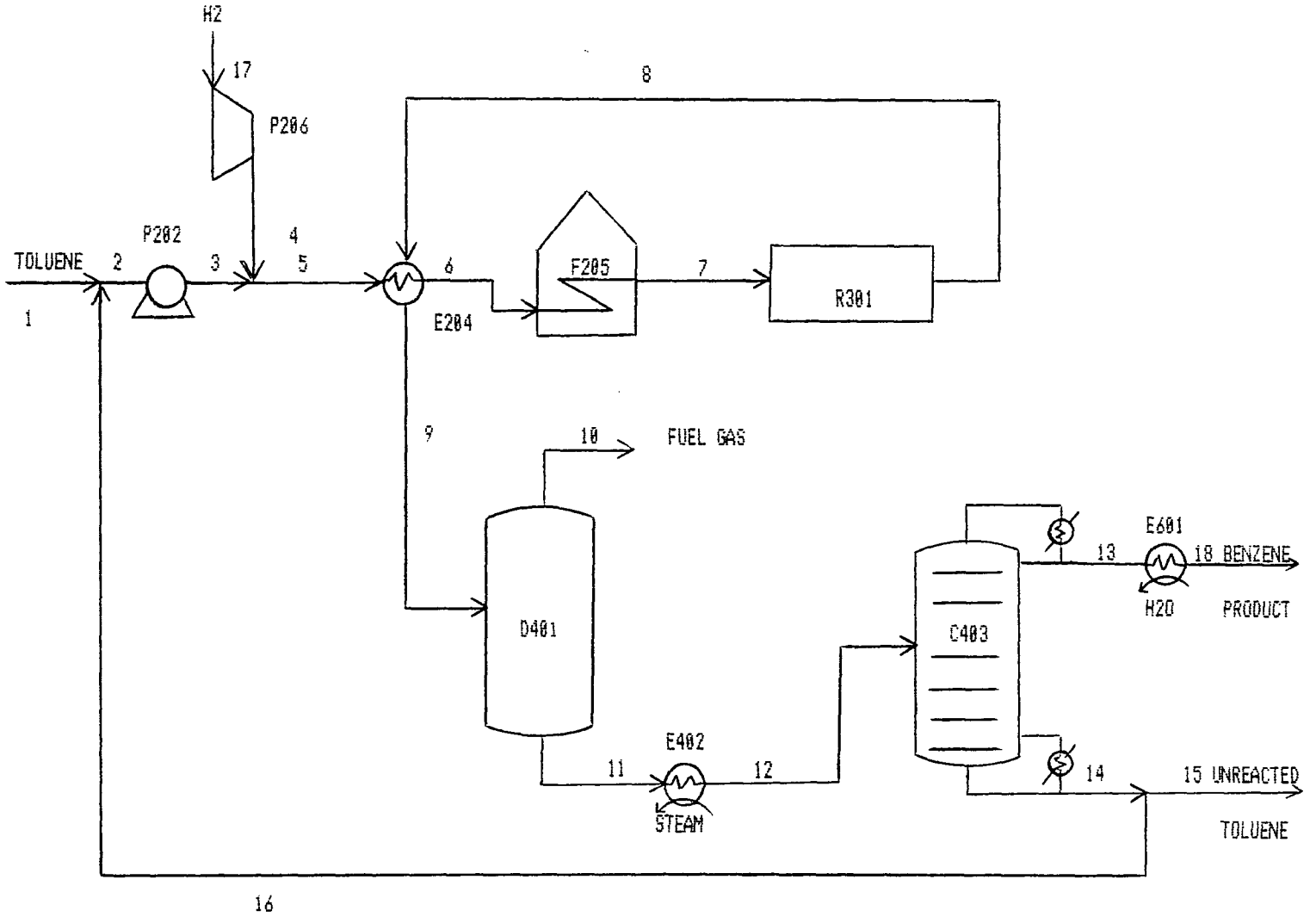


Figure 3.1 Process Flowsheet

Toluene at 70 F and 1 atm is mixed with recycled toluene and compressed to 500 psia. Then, the toluene is mixed with fresh hydrogen, and heated to a reaction temperature of 1040 F through contact with the reactor effluent stream and passage through a fired heater. In the reactor, seventy-five percent of the toluene is converted to benzene and methane gas.

To separate the methane and excess hydrogen, the temperature of the stream is quickly reduced to 40 F, and the pressure reduced to 30 psia in a flash unit. The methane and excess hydrogen leave the system, and the liquid product is heated to 210 F using saturated steam at 250 F.

To separate the benzene from the unreacted toluene, the liquid flash product is distilled. Ninety-nine percent of the benzene fed to the column leaves in the overhead stream. The bottom stream is recycled after a ten-percent purge is made to prevent the buildup of benzene. Only a partial condenser is used with the column in case any hydrogen still remains in the distillate stream. The distillate stream is then cooled to 70 F using cooling water, and then sent to another portion of the plant.

The input stream compositions are given in Table 3.1 along with the initial estimates for streams 8 and 16. Table 3.2 contains a summary of the important characteristics of each unit.

Table 3.1 Initial Stream Data

Stream Name	1	8	16	17
Pressure, psia	14.696	490.0	25.0	100.0
Temperature, R	530.0	1400.0	690.0	530.0
Vapor Fraction	0.0	1.0	0.0	1.0
Composition, lb moles/hr				
Hydrogen	0.0	300.0	1.0	300.0
Toluene	100.0	25.0	22.5	0.0
Benzene	0.0	75.0	1.0	0.0
Nitrogen	0.0	6.0	0.1	6.0
Methane	0.0	75.0	1.0	0.0

Table 3.2 Equipment Characteristics

Unit C403 - Distillation column

99 % of light key and 0.8 % of heavy key in distillate, actual reflux ratio/minimum reflux ratio = 3, light key = benzene, heavy key = toluene, partial condenser

Unit D401 - isothermally flashes a stream and separates the vapor and liquid products

flash temperature 500 R, flash pressure 30 psia

Unit E204 - 3 identical Process-Process heat exchangers

1 shell pass, 1 tube pass, 20 R approach, 5 psia pressure drop for each stream, U = 70

Unit E402 - Single stream heated by steam

1 shell, 1 shell pass, 1 tube pass, 5 psia pressure drop, U= 15, output temperature = 670 R

Unit E601 - Single stream cooled by water

1 shell, 1 shell pass, 1 tube pass, 5 psia pressure drop, U= 90, output temperature = 530 R

Unit F205 - single stream heated by fuel oil combustion

5 psia pressure drop, output temperature 1500 R

Units M201 & M203 - Mix 2 streams to form 1 stream

Unit P202 - Pump - Output pressure 500 psia

Unit P206 - Compressor - 3 stages, output pres. 500 psia

Unit R301 - Reactor

75 % conversion of toluene, $C_7H_8 + H_2 = C_6H_6 + CH_4$

Unit S501 - splits stream 14 into two streams: 15 and 16

10 % of stream 14 will become stream 15
90 % of stream 14 will become stream 16

Leesley (1982) believed that assuring the convergence of all of the components was the major reason that material-recycle loops were considered pitfalls. Often a program cannot tell if a component, which is present in only minute amounts, has converged once the other components have converged. To check for this condition in the example, the unreacted toluene is recycled back to mix with the toluene feed.

Energy recycles were considered by Leesley (1982) to be difficult to handle due to their slow speed of convergence. A process-process heat exchanger, for example, would need to be evaluated twice for each mass iteration. This is even more complicated when the energy recycle loop is added inside a mass recycle loop.

This is the case in the example. If no nesting is done, stream 9 is calculated at the same time stream 6 is. This means stream 9 will not be updated again until the next mass recycle loop iteration, even if stream 8 changes after R301 is calculated. The only solution to this problem is either to nest the energy loop inside the mass recycle loop or to calculate unit E204 twice for each mass recycle iteration, once before F205 is calculated and once after R301 is calculated. The first attempt with the example is to see if the program can converge without any of these special solutions.

Leesley (1982) stated a wide range of operating temperatures, pressures, and components was often a problem for the thermodynamic routines. Many of the correlations are only valid over small ranges. If a simulation covers a wide range of conditions, more rigorous correlations are needed which require longer execution times.

The example chosen attempts to challenge the thermodynamic routines by using a temperature range of 500 R to 1500 R, a pressure range of 14 psia to 500 psia, and a stream with a very high concentration of hydrogen. The accuracy of the correlations is not verified, just whether the program will operate in the given range.

Having a wide range of process units in a simulation can create a problem with the computer memory. This is either aggravated or saved by the loading structure. In the example, gas and liquid pumps, four different types of heat exchangers, a reactor, a flash column, and a plate column are all used. These are the units most commonly encountered in educational design projects.

The differing methods among companies and constant yearly updating make accuracy in cost estimation routines difficult. Only one of the programs in this evaluation could handle costs so this ability is not included in the example.

The ability to add additional components and units demonstrates the flexibility of the program. It is very

important that subroutines and components be able to be added without difficulty. Neither of these are involved in the example also, but the technique of doing this is mentioned.

CHAPTER 4

CHESS

The Chemical Engineering Simulation System, CHESS, was developed under the direction of R. L. Motard in 1968 at the University of Houston. The intended users of the program were educators and students. (J. R. Fair, 1980) Since 1968, two other versions have been released. The program used in this research is the third version, released in January of 1978 (as determined from the program listing).

Abilities

CHESS contains nineteen different unit model routines. These are listed in Table A-1. More models can be added by inserting their code directly into the code for CHESS. This is not a good method for educational users since the entire program must then be recompiled. (The cost of recompiling the program on the CYBER is approximately four dollars in 1984.) It should also be noted that the average student account does not have enough computer memory available to it to allow the CHESS program to be edited.

CHESS has only one rigorous model. That model is for stage-by-stage distillation calculations. The rest of the models use approximations such as temperature approach in heat exchangers.

The design abilities of CHESS are also limited. In the DISC distillation block, for example, only simple specifications such as the fraction of light key to the distillate can be specified. For controlling component ratios, stream temperature, or flow rate, the model CTRL has to be used. This model replaces the unit being controlled, which is limited to the MIXR, ABSR, DISC, and ADBF blocks. No provision has been made for optimizing or controlling entire processes.

CHESS's thermodynamic database contains information for ninety-eight components. Most are compounds used in the petroleum industry. Only seven are not hydrocarbons. A complete list of available components can be found in Table A-2 in Appendix A. Water can only be used as a vapor, except when used as a utility in a heat exchanger. Some common classes of hydrocarbons not included in the database are alcohols, acids, and halogenated hydrocarbons. Those that are included are unsaturated, saturated, paraffinic, naphthenic, and aromatic hydrocarbons.

Using additional components is easy. The required data is simply added in the input where indicated in R. L. Motard and H. M. Lee (1971). Unfortunately, this data cannot be saved from one run to the next. To permanently add a new compound, the data should be inserted into the program directly. Only two permanent additions are possible without rewriting the program, or deleting other compounds.

Deleting other compounds is possible because the name and physical property information for the compounds are stored on data statements. Data statements for other compounds can be switched provided their physical properties can be approximated by the thermodynamic correlations.

A Chao-Seader physical property package as modified by Grayson-Streed is the only correlation included in the package. The range of operating conditions which correspond to that correlation are listed in Appendix A.

Software

When performing the unit calculations, CHESS starts with a unit for which all of the inputs are known and continues until a unit is found for which the inputs are not known. Then, CHESS looks to see if any tearing has been specified by the user.

If no tearing specifications have been made, the program uses a procedure proposed by Barkley and Motard to determine the calculational order for the loop and a tear stream (See R. L. Motard, et. al., 1971). Partitioning information is not included in the input data or in the tearing procedure because it is performed automatically by the program. CHESS does not contain any nesting capabilities. All nested loops are treated as one.

For additional information concerning how CHESS determines the calculational order, an article by F. L.

Worley, Jr. and R. L. Motard (1972) discussed how information was transferred during a simulation program with reference to the CHESS program.

The tearing specifications which can be supplied by the user are the tear stream and the order of calculation of the units in the partition. If one part of the information is left out the other can be determined by the system.

On simple loops, it is easier to input the tearing information manually. This saves execution time because the calculational order finding procedure is avoided. It should be noted that units can be repeated in the calculation loop to allow one unit to be calculated twice in a single calculation loop if needed.

Once the order of calculation is determined, the sequential modular approach is used to perform the unit module calculations. If any recycle streams are present, the Bounded Wegstein method is used to accelerate convergence. A more elaborate convergence routine is not used because the writers believed "a simple convergence scheme, carefully applied, will use less time and memory space than a much more complex scheme which converges in fewer iterations." (Motard, et. al.;1971)

Interface

The input data to CHESS is divided into seven sections: the process configuration, the equipment

parameters, the extensive stream properties, the intensive stream properties, the recycle information, and the thermodynamic constants for any additional compounds. All of these sections but one use the FORTRAN NAMELIST input format.

Originally the data for all of the sections was inputted using NAMELIST format. One section, PMLIST, had to be changed to fixed format because it required the input of character strings. The CYBER can only read character strings as input in FORTRAN 4 when the read is formatted. NAMELIST format input is not formatted.

Due to this change, the input is somewhat harder to learn. The students now have to remember in which columns the variables are to be placed and in which order. The NAMELIST format is only somewhat easier to remember and learn. The students are still required to remember what order the entries are inputted for the arrays, but not which array comes first and which comes second in a section.

In the recycle information section, it would be easier for the students to remember the input data if the variables are easier to associate with their name. For example, KE2 is the recycle unit calculation order, KE3 is an array of tear streams, and KE4 is an array of the streams to be used in the convergence acceleration.

Another disadvantage the NAMELIST format has is the requirement of remembering the dimensions of the vectors.

The number of elements given in the input data must match the dimension of the array or the program will stop.

Most of these disadvantages to both types of input can be solved by a summary sheet of the input data. For the fixed format data, the variable order and their respective column positions should be listed. For the NAMELIST data, the variables should be listed for each of the sections and along with their dimensions. A similar sheet appears in Table A-4 in Appendix A of this report.

The output which CHESS provides is an echo of the input data and a listing of the output results for the stream connections, stream compositions, and equipment parameters. This can occupy a minimum of fourteen pages.

The ability to get additional output is important because it allows the students the opportunity to determine for themselves why their simulations are not converging. To get additional output, three different variables may be set: KTRACE, PSTRMS, and NPFREQ. KTRACE will allow a listing of the models called during the simulation to be made, as well as, the input stream information for each block being calculated.

During one example which contained 10 units and took 4 iterations to converge, the output increased from 18 pages to 20 pages, the execution time increased from 2.230 sec to 2.277 sec, and the printing costs rose from $ 0.48 to $ 0.52

when KTRACE was set. These numbers would have increased if more iterations were required to converge the recycle loop.

NPFREQ allows an intermediate stream matrix summary to be printed. PSTRMS is the interval for a printout of the equipment summary. PSTRMS lengthens the output approximately 2 pages for each interval printed. An equipment summary is not very informative for checking errors since no stream information is given.

CHESS also provides error messages to monitor convergence. The most common error message occurs during the flash routines when the temperature is out of range. Normally, this message is only informative.

For example, if the phase of a stream at a given temperature is needed, the flash routines first try to determine what the dew point and bubble point temperatures are. Then, if the actual temperature is between the two, the program tries to determine how much of the stream is liquid and how much is vapor. Unfortunately, if the stream contains hydrogen or other light gases, the dew point temperature can be out of range of the program. This results in an error message but not a fatal condition since the program would then assume that the stream is two-phase which it probably is.

Unfortunately, the program does not indicate when input errors are made. During one example, temperature and vapor fraction were inadvertently switched. The program did

realize vapor fraction could not be greater than one, but it did not print an error message indicating that the input was impossible.

During another example, a stream in a process was supposed to be cooled to a certain temperature by cooling water. During the calculations, though, the temperature of the stream turned out to be lower than that of the cooling water and the desired outlet. Instead of printing a message indicating that the unit would not work, the temperature of the stream was set to the desired value, and a negative area calculated.

The documentation for CHESS has several problems. Although the program code states that it is version three, it's abilities are not the same as those described in the third edition of the manual. Four models have been added, as well as, more options for the ADBF model.

The manual also misleads the student with regard to what stream parameters are required. When looking in the manual, Figure 4.1 is found for the stream parameters. Not all of these are required. Once the stream composition is given, only the stream name, stream type, pressure, and temperature or vapor fraction is needed.

The manual is also misleading in not stating that the equipment node numbers have to be serial. Note the increased confusion in the output of Example 1 in Appendix B. Eventually, the last two units specified in the input

SINJ = J, F, 0., VF, T, P, H, μ, K, Z_L, Z_V,

where

F = Stream flag; 0. = intermediate stream,

1. = feed stream,

2. = product stream,

VF = Vapor fraction, moles vapor/ total moles,

T = Temperature, R,

P = Pressure, psia,

H = Total heat content, BTU/hour,

μ = Viscosity,

k = Thermal conductivity,

z_1 = Compressibility factor for the liquid phase,

and

z_v = Compressibility factor for the vapor phase.

Figure 4.1 Input Stream Parameters

data are dropped. This was seemingly due to the fact that there were twelve units given in the input so the program only considered units which were numbered twelve or under. The actual input which is needed is also given in Appendix B together with the new output. In the manual, this same example is given and the equipment numbers are not serial.

To solve these problems, the manual was revised and is presented in Appendix A. The manual was also reorganized to allow the model descriptions to appear by the input data specifications, and shortened to cut down on the cost of distributing the manual to students.

Structure

CHESS is based on a fixed executive structure. The maximum size of process which it can simulate contains 100 streams, 50 equipment units, 20 components, and 7 streams per unit. This large capacity is good for industry, but only one-half of the capacity dimensioned for would probably be needed for educational purposes. This means that the program requires more memory than that which is actually needed.

The way the CHESS program is loaded also increases the memory requirement. When CHESS first arrived, it was designed to simply load entirely into memory and then to execute. This unfortunately requires too much memory for the CYBER.

To solve this problem, the program has been rewritten in overlay structure. This is shown in Figure 4.2. The main overlay contains the main program, the flash routines, and the thermodynamic correlations. It remains in memory the entire time the program executing.

The primary overlays are only loaded as they are needed, and only one is loaded into memory at any given time. Therefore, each primary overlay must perform a different function. Overlay (1,0) reads the input data, overlay (2,0) contains the tearing and calculational order finding routines, overlay (3,0) sets up the data for the thermodynamic correlations, overlay (4,0) prints the output results, and overlay (5,0) calls the unit operations subroutines.

The secondary overlays can only be called by the fifth primary overlay. As with the primary overlays, only one secondary overlay can reside in memory at any given time, and they can only be loaded when they are called. The secondary overlays each represent a unit module routine.

With these modifications, CHESS is able to load into memory and execute without having memory problems. Now, the maximum memory required to execute the program is the memory needed by the main overlay, the fifth primary overlay, and the longest unit model routine. This is approximately 160 blocks.

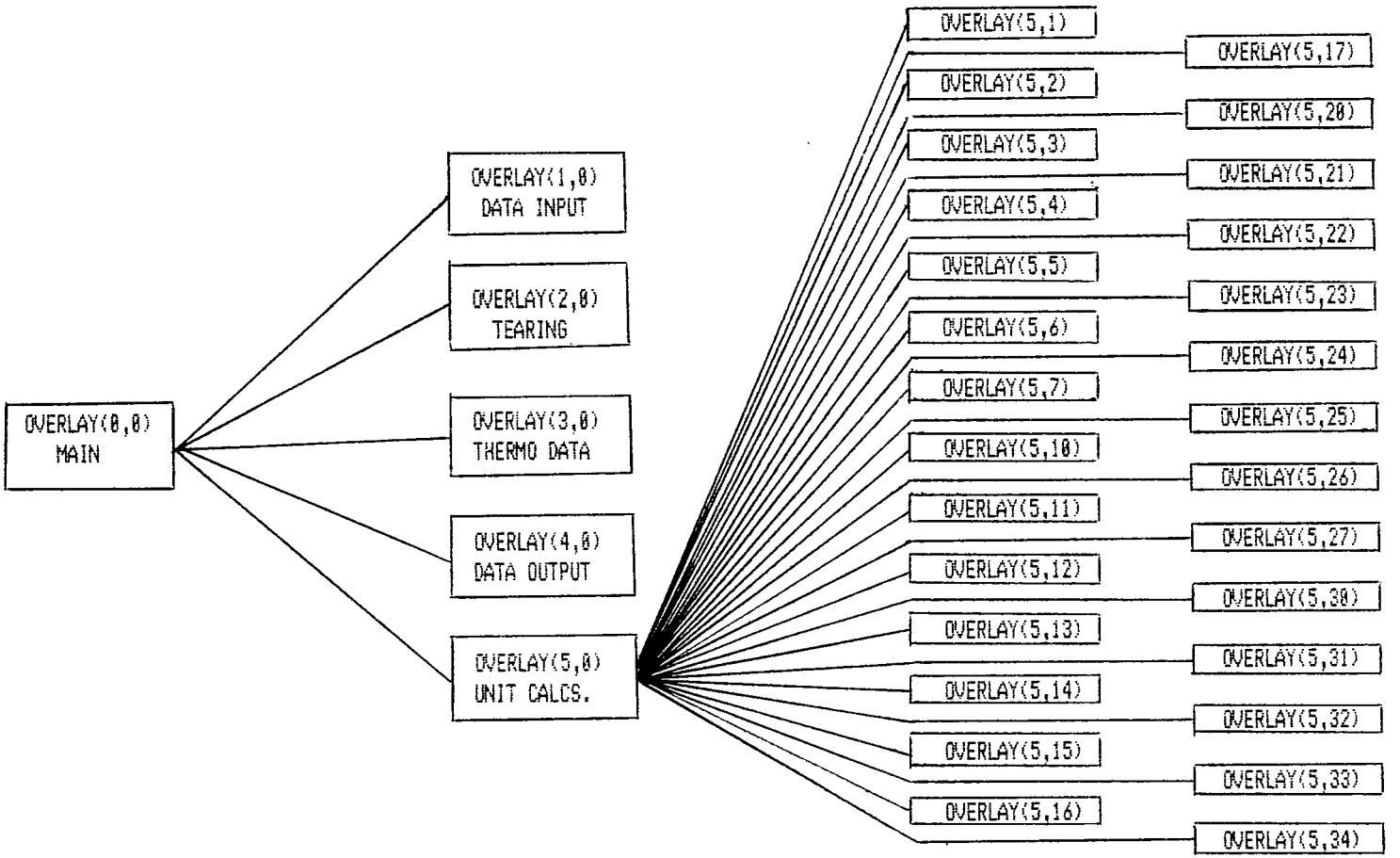


Figure 4.2 CHESS Overlay Structure

Some changes which need to be made to accommodate the switch to CDC FORTRAN 4 from IBM FORTRAN 4 are hollerith input instead of single quotes for character strings, similar arguments between the main entry point and any subsequent entry points in subroutines, the command DOUBLE PRECISION instead of REAL*8, the NAMELIST commands after all of the character definition, DIMENSION, and BLOCK statements, and EQUIVALENCE (EQPAR(1,1),EQP1), (EQPAR(2,1),EQP2), etc. instead of just (EQPAR(1,1),EQP1).

Once these modifications to CHESS were completed, the program was saved onto a tape through a DEC 10 computer. To install the program, it needs to be restored to a DEC account and sent to the CYBER to be compiled. These procedures are listed in Appendix A. This procedure is very simple and inexpensive. The cost of installing the program is four dollars, and the cost of storing the program is thirty-two cents per day. The program compiles in 20 cpu seconds, and occupies 971 blocks of permanent storage.

Executing the program is also easy and inexpensive. The average simulation costs two dollars and takes nine cpu seconds. To allow for more complicated flowsheets, the student account numbers should have a 12 second time limit.

Control blocks greatly increase the execution time required by the programs. In Example 2 in Appendix B, the execution time is 33 cp seconds for only a controller and a stripper. The cost for the example is $2.84.

Example

CHESS handles the toluene dealkylation example very well. Very few problems are encountered. The execution time is 5.3 cp sec, the memory required is 160 blocks, and the cost is $1.27. The input and output are given in Appendix A. Figure A-1 shows the process information flowsheet for the CHESS program.

When handling the material recycle loops, the errors in the material balances are always less than 1 percent. The only units which contain errors in their material balances are the DVDR and MIXR blocks.

The calculation order finding routines conclude that the units in the loop should be solved in the order: 6, 7, 8, 9, 10, 11, 1, 2, 4, 5. This does not seem like the best order, though, because the inputs are not known for unit 6. In fact, the only unit in the loop for which the inputs are known or given an initial estimate of is unit 1. This means that during the first iteration, units 6 through 11 will not be evaluated. The half of the first iteration is useless.

The order was, therefore, inputted as 1, 2, 4, 5, 6, 7, 5, 8, 9, 10, 11. This order allows all of the units to be evaluated during the first loop, and stream 9 to be updated after stream 8 was updated also. The number of iterations the program needs to converge is only four indicating that convergence of the energy and mass recycle loops is quick.

Originally the range of temperatures was from 500 R to 1500 R. 1500 R is the maximum temperature the thermodynamic routines allow, so the upper range was lowered to 1400 R to assure that the temperature is always less than 1500 R.

As expected, the presence of hydrogen did cause out-of-range temperature limits to occur when the dew point was calculated. These were not fatal. No problems occurred due to the pressure range of 14 to 500 psia.

Figure A-1 shows which units can be represented by which models. Altogether, eight models are needed to simulate the twelve different units. CHESS is able to simulate all of the units but one. To heat a stream with a heating medium other than fuel oil, the heating medium must be specified as a separate component, and streams must be specified which are composed of only that component. This is avoided in unit 9 by using an option in the HXER model which just determines the heat lost for the stream given it's outlet temperature.

CHAPTER 5

PROCESS

The PROCESS simulation system is the property of Simulation Sciences, Inc. It was released in January of 1979 (See PROCESS Simulation Program Input Manual, 1983). The program is available to universities through written permission from Simulation Sciences, Inc. in Fullerton, California, but the intended users of the system are process engineers in the chemical, petroleum and synthetic fuels industries. The program used in this comparison is the third version released in September 1983.

The PROCESS simulation system is composed of two different programs, PREPRO and PROCESS. PREPRO converts the free format input into a fixed format sequential data file (PROCESS Installation Guide for CDC NOS Systems, 1984) which is then inputted to the PROCESS program which performs the actual simulation.

Abilities

The PROCESS system contains 28 different models. These are listed in Table 5.1. The models range from very general to very specific. The addition of other models is possible but not needed for educational purposes, they

Table 5.1 PROCESS Models

COLUMN	Rigorous Distillation
SHORTCUT	Shortcut Distillation
FLASH	Flash Operations
MIXER	Mixes up to six streams
SPLITTER	Splits a stream into six possible streams
VALVE	Valve
HX	Heat Exchanger
COMPRESSOR	Compressor
EXPANDER	Single Stage Isentropic Expansion
PUMP/PWRT	Pump or Power Recovery Turbine
SEPARATOR	Component by Component Split
REACTOR	Reactor
LNGHX	Heat Exchanger for Multiple Hot or Cold Streams
F3PHASE	Three Phase Flash Operations
SHIFT	Shift Reactor
METHANATOR	Methanation Reactor
LLEXTRACTOR	Liquid-Liquid Extraction
BATCH	Batch Distillation
CSTR	Continuous Stirred Tank Reactor
DEPRESSURE	Depressuring Unit
ELDISTILLATION	Electrolytic Distillation
GIBBS	Gibbs Reactor
EXDISTIL	Extractive Distillation

Table 5.1 Continued

CALCULATOR	Calculator
CONTROLLER	Feedback Controller
MVC	Multivariable Controller
SET	Feed Forward Controller
OPTIMIZER	Flowsheet Optimizer

require obtaining additional information from Simulation Sciences.

PROCESS contains both rigorous and simple models. These are not always different models but options which can be chosen. An example is the COLUMN model. The COLUMN model can be a simple tray distillation column with a total condenser and reboiler, or a column with side heaters, pump-arounds, and a choice of tray designs and spacings.

The models have very extensive specification abilities for the output streams. Specifications may be made on the rate, purity, temperature, pressure, and physical properties. These specifications can also be relative to another stream, in addition to, absolute amounts. This specification ability makes the PROCESS program very good for design mode calculations.

To control process units, both feedback, multi-variable, and feedforward models are included in addition to the specification abilities. A flowsheet optimizer is also included in the system to optimize entire processes, but only certain variables can be optimized.

The component database in PROCESS is very extensive. It contains over 980 compounds for use in both hydrocarbon and chemical industries. The classes of hydrocarbons included in the database are acids; alcohols; aldehydes; amides; amines; ethers; esters; keytones; nitrogen, sulfur,

and halogen derivatives; and paraffinic, naphthenic, unsaturated, saturated, and aromatic hydrocarbons.

There should be no reason why more compounds would need to be added if the program is used for educational purposes. If additional compounds are needed, their information can be saved permanently in a private data file or only temporarily.

The PROCESS package contains approximately 13 different thermodynamic correlations. Only an approximate number is given because different correlation methods may be used for different units in a simulation. The systems for which the correlations apply are heavy hydrocarbon, synfuels, light hydrocarbon, and natural gas. The range of operating conditions varies from 0 to 3000 psia and -225 F to 500 F depending on the correlation.

Software

When determining the order of calculation of the units, the order which the units are given in the input is the order in which they are calculated. To change this order without rearranging the data, a SEQUENCE card must be used. Similarly to CHESS, partitioning is automatic. Tearing and nesting can either be determined by the program or inputted manually. If nesting is performed automatically by the program, the entire partition is assumed to be a single loop.

When specified, the loops can be nested by giving the first unit in the loop and the last unit in the loop. Then, all of the units inputted between the two are part of the loop regardless of whether or not they actually are. Thus, the students are required to consider what order the units are given in the input.

The sequential modular approach is used in the unit module calculations. To promote recycle convergence, three different approaches are possible: direct replacement, Bounded Wegstein, and Dominant Eigenvalue. Direct replacement is the default method.

The dominant eigenvalue method is different than that proposed in Chapter 2. The new estimate for the recycle rate is

$$X_{k+1} = X_{k-1} + \text{ALPHA} * (X_k - X_{k-1})/(1-\text{LAMBDA}) \qquad (5.1)$$

where ALPHA is a damping factor and

$$\text{LAMBDA} = \Delta X_k \ / \ \Delta X_{k-1}. \qquad (5.2)$$

These convergence methods are more flexible than those used in CHESS and FLOWTRAN because the limits of the accelerations may be set by the user.

Interface

The PROCESS system uses free format input with keywords to input the data. This requires a preprocessor,

but greatly increases the student's ability to remember and learn the input. For example, to specify the output temperature for a stream, the student only needs to insert

SPECIFICATION STREAM=stream number,TEMPERATURE=temp

in the input.

The input is also easier to remember since the input is consistent from unit to unit. The pressure is always inputted using PRESSURE, the temperature using TEMPERATURE, the rate using RATE, etc.

One feature of the interface which is very nice is the ability to input the data and output the results in several different units. This saves the engineer time if the output is desired in several different units, or if the output units are to be different from the input units.

The output from the program is very extensive. I have never seen an entire output from PROCESS, but I expect it is very long. This impression is obtained by knowing that a directory of the output is optional, and in the PROCESS manual, an output is given which starts on page 21 and continues for 19 pages.

The output is long because of the unit summaries. The output for each unit is on a separate page, and only four stream compositions are given on a page. This type of output is good for final design reports in industry but is far to extensive to outweigh the additional cost for

education. The length of output should be able to be shortened.

An example of the output from the COLUMN block is given in Figure 5.1. For each stage in the column, the temperature, pressure, and liquid and vapor flowrates are given.

Intermediate calculation results are possible for rigorous distillation, shortcut distillation, controllers, and recycle loops. For the recycle loops, the component molal rates of all the streams involved in the loops can be printed after each trial. This will be helpful for the students when trying to diagnose errors.

The documentation for the PROCESS package is not intended to be distributed to a large number of people. As such, it is very lengthy. This makes distributing the manual to students very expensive. Shorter input guides are available but they are only summaries (See PROCESS Key Word Input Guide, 1983). To use the shorter guides, the students would need to have the input explained first.

It was also noted while looking at the code for the program that some of the thermodynamic correlations mentioned in the manual are not actually available in the version distributed to universities. Simulation Sciences probably charges extra for these options but nothing is mentioned about this in the input manual. This means that the manual is wrong in mentioning these options as being

```
VERSION 0783                                SM
SIMULATION SCIENCES, INC.                PROCESS                    PAGE  31
PROJECT  USER GUIDE                   UNIT 10 - T-1                 SIMSCI STAFF
PROBLEM  CH PLANT                       SOLUTION                    JAN 1981

          I SUMMARY FOR COLUMN UNIT 10 - T-1 ,

 1 TOTAL NUMBER OF ITERATIONS
       FAST METHOD                                                      0
       SURE METHOD                                                      5

 2 COLUMN SUMMARY

 TRAY    TEMP   PRESSURE      NET FLOW RATES, LB MOLS/HR        HEAT(COOL)ER
                              LIQUID    VAPOR                     DUTIES
         DEG F     PSIG     PHASE(L)  PHASE(V)   FEED    PRODUCT  MM BTU /HR

   1     27.3    215.00       11.3                        10.1V     -0.2797
   2    333.3    220.00       22.4      21.4
   3    364.9    220.00       25.7      32.5
   4    370.4    220.00       26.4      35.8
   5    371.4    220.00       26.5      36.5
   6    371.6    220.00       26.5      36.6
   7    371.6    220.00       26.5      36.7
   8    371.6    220.00      571.0      36.7   375.4M
   9    411.9    220.00      710.1     205.7
  10    416.2    220.00      728.9     344.8
  11    416.6    220.00      731.0     363.6
  12    416.7    220.00      731.2     365.7
  13    416.7    220.00      731.2     365.9
  14    416.7    220.00      731.2     366.0
  15    416.7    220.00                366.0              365.3L     3.0668

 3 FEED AND PRODUCT STREAMS                          MASS RATES   HEAT RATES
    * FEED STREAMS:                                  LB MOLS/HR   MM BTU /HR
   14   TO TRAY 8 IS MIXED  FROM UNIT 11, E-4     0.37537E 03  0.36191E 01
    * PRODUCT STREAMS:
   16   IS  LIQUID STREAM FROM TRAY 15             0.36526E 03  0.64068E 01
   15   IS  VAPOR  STREAM FROM TRAY  1             0.10110E 02 -0.55581E-03

   OVERALL MASS BALANCE, (FEEDS - PRODS)          0.00000E 00
   OVERALL HEAT BALANCE, (HIN - HOUT)                          0.19073E-04

 4 SPECIFICATION VALUES

   PARAMETER TRAY COMP. SPECIFICATION  SPECIFIED    CALCULATED
     TYPE     NO   NO       TYPE        VALUE        VALUE
   STRM 15     1    4    ML RATE RATIO 0.1000E-03  0.1000E-03
   TRAY LIQD   1         ML RATE RATIO 0.3000E-01  0.3000E-01
```

Figure 5.1 COLUMN Block Output

available. There are probably other options mentioned in the manual which are not actually available, but this is not known.

Structure

The executive program for PROCESS is fixed. The largest process which the program is able to handle consists of 50 components, 150 streams, and 75 units. This is quite wasteful considering the average student simulation would probably only use one third of this capacity.

Four files need to be copied from the tape to install the PROCESS package. These are SEGPRO, PREPPL, PROCPL, and PROLIB. The SEGPRO file contains the directives or instructions for how the PROCESS program is to be segmented, PREPPL and PROCPL are coded files containing the program listing for the PREPRO and PROCESS programs, respectively, and PROLIB is a library containing the physical property correlation constants for the component database. The commands for installing these programs are given in Appendix C.

PROCESS and PREPRO are coded before copying onto the tape to allow the programs to fit onto one 1600 BPI, 2200 foot tape. This coding is produced by sending the program code through a utility called UPDATE. The UPDATE utility removes blank spaces from lines, allows similar lines in a

program to be written only once onto a tape, and numbers each line for easy reference.

If the program needs corrections, new code can be added or old code deleted just by referencing the line number. This works out very well since some internal functions referenced in the program are not available on the CYBER, and their references must be deleted without editing the program on the DEC 10.

Once decoded, the PREPRO program is small enough that it can be installed without any segmentation or overlaying to reduce it's memory requirements. PREPRO is simply compiled and loaded into memory in it's entirety.

The PROCESS program, on the other hand, is very large, and segmentation is required. A segment is similar to an overlay except each segment does not have to be a separate program. Unfortunately, segmentation does not reduce the memory requirement enough for PROCESS to allow it to execute.

Altogether, $42.56 (in 1984 for the U of A CYBER) is needed to install the PROCESS package. This installation cost does not include the cost of loading the PROCESS program since it will not load correctly, but does include the compilation costs. A more detailed breakdown of the costs is shown in Figure 5.2.

If the PROCESS program had been able to load, the approximate cost of storing the program on the CYBER would

Procedure	Cost,$
Unloading the program from tape	1.64
Correcting and running UPDATE on PREPRO	0.72
Running UPDATE on PROCESS	9.39
Compiling and loading PREPRO	3.35
Compiling PROCESS	27.46

Figure 5.2 CYBER Installation Costs For PROCESS

be approximately six dollars per day. This could become very expensive over long periods of time.

Example

The example, of course, was never attempted with the PROCESS package, but the input for the example was determined to check the PROCESS simulation package's range of operating units. This input is shown in Appendix D. The process information flowsheet is shown in Figure 5.3.

Unit number 5 was originally meant to be 3 identical heat exchangers. This was solved in CHESS by specifying a single heat exchanger with 3 shells. PROCESS does not have this option which means that either three separate heat exchangers will have to be specified, or the three will have to be combined into one. For this simulation, the three identical heat exchangers were treated as one.

The PROCESS program is also unable to model a fired heater. This can be solved by specifying the fuel as a separate component, combusting a stream of that component in a reactor, and then setting the heat duties of the two units equal. The problem with this approach is that the amount of fuel needed in the heater will have to be known in advance since the outlet temperature of a heat exchanger cannot be specified in advance once the duty is specified. This unit was subsequently replaced with a steam heater.

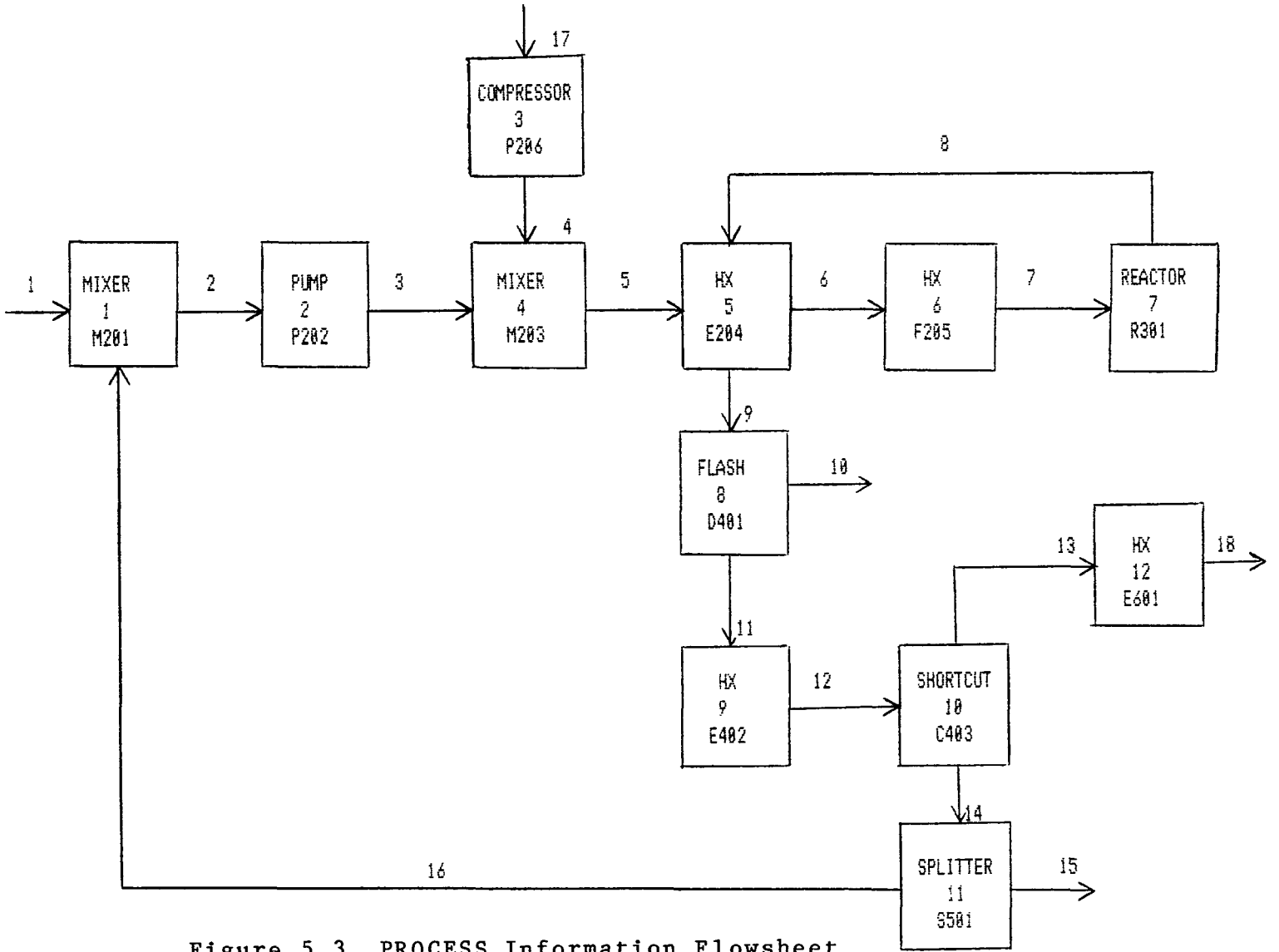


Figure 5.3 PROCESS Information Flowsheet

CHAPTER 6

FLOWTRAN

The FLOWTRAN program was developed by Monsanto Co., in St. Louis, Mo. Stanley I. Proctor (1983) gave the history of FLOWTRAN as the following. Initial work began in 1964 on the physical properties, input/output routines, and general system design. This was completed in 1966. In 1969, the system was marketed outside of the company using a service bureau. This was discontinued in 1973 since the system was offered for license in 1972.

Educators were first able to use FLOWTRAN in 1974 when the system was made available to American and Canadian universities through the CACHE committee of the American Institute of Chemical Engineers and a service bureau. To reduce the cost of using the FLOWTRAN program, load modules were made available to universities in 1982 to use on their own computers.

The FLOWTRAN system consists of the files INF, PREPRO, VLE, PROPTY, FTSLIB, FTPUBF, FTPRIF, and FLOEXE. The INF file is a program which acts as an interface between the thermodynamic data files, FTPUBF and FTPRIF, and the other FLOWTRAN programs; the VLE program converts raw phase equilibrium data to liquid activity coefficients for use in

the thermodynamic correlations; PROPTY is similar to VLE except it converts raw property data to physical property constants; FLOEXE is a file containing the commands to execute the program; PREPRO translates the input data into an executive program; and FTSLIB is a library of compiled FLOWTRAN routines.

Two different versions are used in this research. One is for a CYBER 175 with the NOS operating system and another is for a VAX 11/750 with the VAX VMS operating system. The CYBER version comes from the University of Washington where they modified the program for a CYBER with a NOS operating system. The VAX version is distributed by the CACHE committee.

Abilities

The FLOWTRAN system contains 60 different models. Of those 60, 20 are costing blocks. These are listed in Table 6.1. Additional models can either be added to a library or used only once. FLOWTRAN also has the unique ability to evaluate any number of FORTRAN statements between unit simulations. This could be very useful.

The FLOWTRAN system contains a large number of models because each model simulates only one unit. The pump unit model is separate from the compressor unit model, each type of flash is a separate model, and each unit also has a separate cost block, for example. The rigorous models which

Table 6.1 FLOWTRAN Models

IFLSH	Isothermal Flash
AFLSH	Adiabatic Flash
BFLSH	General Purpose Flash
KFLSH	Isothermal Three Phase Flash
FLSH3	Adiabatic/Isothermal Three Phase Flash
FRAKB	Rigorous Distillation
DISTL, DSTWU	Shortcut Distillation
SEPR	Constant Split Fraction Separation
AFRAC	Rigorous Distillation/Absorption
ABSBR	Rigorous Absorber/Stripper
EXTRC	Rigorous Liquid-Liquid Extraction
EXCH1, EXCH3	Shortcut Heat Exchanger
CLCN1	Shortcut Cooler Condenser
DESUP	Shortcut Desuperheater
HEATR	Heat Requirements
EXCH2	Shortcut Partial/Total Vaporizer/Condenser
BOILR	Shortcut Reboiler/Intercooler
HTR3	Three Phase Heater/Cooler
ADD, MIX	Stream Addition
SPLIT, PART	Stream Split
PUMP	Centrifugal Pump Size and Power
MULPY	Stream Multiplication by a Parameter
GCOMP	Compressor and Turbine
SCVW	Bounded Wegstein Stream Convergence
CNTRL	Feedback Controller

Table 6.1. continued

PCVB	Multiple-Parameter Control Block
DSPLT	Distillate Feed Forward Control
RCNTL	Ratio, Sum, and Difference Feedback Controller
CAFLH, CFLH3, CIFLH	Flash Drum Cost
CAFRC, CDSTL, CFRKB	Distillation Column Cost
CABSR	Packed Absorber Cost
CCLN1, CEXC1, CEXC2, CEXC3	Heat Exchanger Cost
CPUMP	Pump Cost
CCOMP	Compressor Cost
CTABS	Tray Absorber Cost
CHETR	Heat Exchanger Cost
BPROD	By-Product Stream Value
PRODT	Product Stream Value
RAWMT	Raw Material Stream Value
PROFT	Profitability Analysis
SUMRY	Stream Output Editor
TABLE	Component Physical Properties Table
GAMX	Liquid-Activity-Coefficients Table
SPRNT	Stream Print Block
ASTM	Analytical Distillation of a Stream
CURVE	Heating and Cooling Curves
REACT, XTNT	Chemical Reactor
AREAC	Adiabatic Add/Subtract Reactor

the FLOWTRAN system contains simulate stage by stage distillation and absorption.

The costing abilities of FLOWTRAN include costing routines for individual blocks, as well as entire processes. This is very useful. To conserve execution time, the costing blocks are not evaluated until the simulation has converged.

The design abilities of FLOWTRAN are very similar to CHESS's abilities. The usual design specifications and operating conditions are all which need, and are able, to be specified with the exception of tray efficiency in the rigorous distillation model.

The controllers which the program contains are very good. They are able to optimize on any parameter which can be given as input. The actual types of controllers are: feedback; multiple parameter; and ratio, sum, and difference feedback. This means control parameters can be made on both absolute and relative amounts. Any parameter may also be varied to perform repeated simulations without respecifing the entire input by using the case study ability.

FLOWTRAN has 180 components in it's database. Most of these are common compounds with less than seven carbon atoms per molecule. The classes of compounds supported are acids, alcohols, aldehydes, amines, aromatics, ethers, esters, keytones, and halogenated, naphthenic, paraffinic, saturated, and unsaturated hydrocarbons.

It is possible to add new compounds as well as to create a private database. The programs VLE and PROPTY make this addition especially easy since they automatically compute the constants which are needed for the correlations given specific physical properties.

Unlike CHESS and PROCESS, the FLOWTRAN system allows the specification of separate vapor pressure, vapor fugacity, liquid fugacity, and liquid activity estimation procedures. The range of operating conditions which these correlations apply to is not given in the documentation (J. D. Seader, W. D. Seider, and A. C. Pauls; 1977).

Software

The FLOWTRAN system cannot determine the order of calculation of the units in a simulation. It relies entirely on the order the units are given in the input. If the process is cyclic, a special, SCVW, block must be added. This block indicates which stream is to be cut and which unit in the input is the first unit in the loop. The program will then recalculate all of the units listed in the input between the first unit and the SCVW block. To nest loops, each loop must be indicated by a separate SCVW block. In this case, each inner loop must converge completely for each trial of an outer loop.

The approach the program uses to perform the unit module calculations is the sequential modular approach. The

Bounded Wegstein method is used to accelerate recycle convergence. The old value, the new value, and the difference between the two are given in the history of each simulation to allow the user to follow the convergence accelerations.

Interface

The data for FLOWTRAN is inputted using free format with keywords. This requires the use of a preprocessor. Fortunately, the preprocessor is very quick. It takes approximately 30 seconds on the VAX. Unfortunately, the keywords are not easily associated with the entries when inputting the process configuration and the equipment parameters. In these entries, the input must follow a certain order after the keyword. This decreases the ability of the students to remember the input. When using the VAX, the input must be given in capital letters.

The output from the program is very short. The minimum number of pages which are printed is eight. This amount is inexpensive to print but not easy to understand. This is because the equipment summaries are vague. Usually they only echo the input data. An example of an equipment output is given in Figure 6.1.

The ability of FLOWTRAN to print additional output is limited. Stream compositions can be printed only after the simulation is complete. If the simulation does not

C403 (DSTWU) FEED=S12 OVD=S13 BOT=S14

HVY KEY COMP. NO.	2.000	LT. KEY COMP. NO.	3.000
SPLIT FOR LT. KEY	99.000	SPLIT FOR HVY KEY	124.000
QUALITY OF FEED	0.000		
DESIRED REFLUX	-3.000	DESIRED NO OF STGS	0.000
TOP PRESS. PSIA	25.000	BOTTOM PRESS. PSIA	25.000
CONDENSER TYPE	1.000		

Figure 6.1 DSTWU Model Output

converge, the program will print the stream compositions during the final iteration, and not for every iteration. A calculational history for each unit is printed for every trial.

The documentation for FLOWTRAN is very good. This is because a book was written especially for teaching students and other engineers how to use FLOWTRAN (J. D. Seader, et. al.; 1977). This book greatly reduces the problem mentioned earlier of the ability to remember the input. The cost of distributing the manual is very low. The book can be purchased for only $ 16.10 in 1985.

Structure

The FLOWTRAN system is based on a very unique variable executive program. The executive program is written by a preprocessor from the input data for each simulation. This reduces the amount of memory the simulation requires since the size of the data storage arrays are set to the size needed for that particular simulation. The maximum capacity of the program is not stated. It depends on the memory contained in the computer and not the dimensions of the executive program.

To install the program on the CYBER, several steps have to be followed. These are listed in Appendix E. These are not the same as those sent with the program. Some changes have to be made to accommodate a NOS/BE operating

system. These changes include altering the format of the tape from internal format to system internal format, converting a NOS library to a NOS/BE library, and not saving the FTPID scratch file from one trial to the next. The cost of installing the program is five dollars. This cost does not include compiling the program because it is already compiled.

The commands also had to be changed to execute the program. Unfortunately, FLOWTRAN is unable to work on our CYBER. It appears that the program cannot read the thermodynamic database. This is probably due to the change in operating systems. If a printout of the database could be obtained, it might be possible to create a new database. This would depend on whether or not the program has trouble writing to a database, as well. If the problem is resolved, the cost of storing the program on the CYBER will be approximately $ 3.20 per day.

The installation of FLOWTRAN on the VAX is simpler. These commands are given in Appendix E. The commands are very similar to those sent with the program except the device name is changed. Once installed, the files in Table 6.2 will appear in the directory. The cost of installing the program is approximately $ 1.50. The total memory requirement for storing the program is 2166 blocks at a cost of $ 1.52 per day.

Table 6.2. FLOWTRAN VAX Files

File Name	Contents
FT.COM	Commands to execute FLOWTRAN
FTBT.FIL	Block table used to store information
FTLB.OLB	Library of models
FTPRI.DAT	Private physical property file named MONSAN
FTPRII.COM	Commands to create a private physical property file
FTPRII.FOR	FORTRAN program to create a private physical property file
FTPUB.DAT	Public physical property file
INF.COM	Commands to execute the INF program
INF.EXE	Compiled INF program
INPUTD.OBJ	Block data subprogram
PREPRO.EXE	Compiled PREPRO program
PROPTY.EXE	Compiled PROPTY program
PROPTY.COM	Commands to execute the PROPTY program
VLE.EXE	Compiled VLE program
VLE.COM	Commands to execute the VLE program

Unlike the CYBER, the execution of FLOWTRAN on the VAX is flawless. Some problems might arise if an execution attempt is made from an account other than the one in which the program is stored. To execute any of the programs in the FLOWTRAN package, it is necessary to create a file containing the data, and then type "@", the program name, and the data file name. The FLOWTRAN program name has been shortened to FT. For example, to execute FLOWTRAN using the data in the file TEST01.DAT, type "@FT TEST01.DAT".

The average cost for a simulation consiting of five units is $ 1.40, and the average time of execution is one minute. It is very difficult to measure the exact execution time and cost of execution of a program on the VAX because of the interactive operating system. The cost of executing a program also includes logging on and off the computer, as well as the connect time for the execution of the program and the input of the data.

Example

The example process works very well on the VAX version of FLOWTRAN. The approximate cost of execution is $ 2.40. The execution time is not measured, but it is close to the average of one minute. The input data and output results are given in Appendix F.

The convergence of the material recycle loop in the example process is 0.0647 lb moles for a relative difference

of 0.2 percent. This occurs after four iterations. The errors in the balance occur while simulating the process-process heat exchanger, E204, and the convergence block, CON1.

The calculation order for the loop is M201, P206, M203, E204, F205, R301, D401, E402, C403, S501, CON1, and E601. The vapor pump is included in the recycle loop to show that the program does not differentiate between the units in the loop and those outside of the loop.

Neither the range of operating temperatures, operating pressures, or compounds used creates problems for the program. This may be because the program has no problems, or it may be because the FLOWTRAN program does not have extensive error message capabilities. The results are similar to those obtained with CHESS which contained several error messages.

FLOWTRAN requires ten models to simulate the thirteen units. The FLOWTRAN information flowsheet for the example process is given in Figure 6.2. The SCVW block is added to indicate a recycle stream.

The HEATR block is used for the fired heater, steam heater, and water cooler because FLOWTRAN does not have the ability to heat or cool a stream with a utility stream. To use water, steam, or fuel gas, a special stream must be inputted composed of only that component, and a process-process heat exchanger model used. This was mentioned in

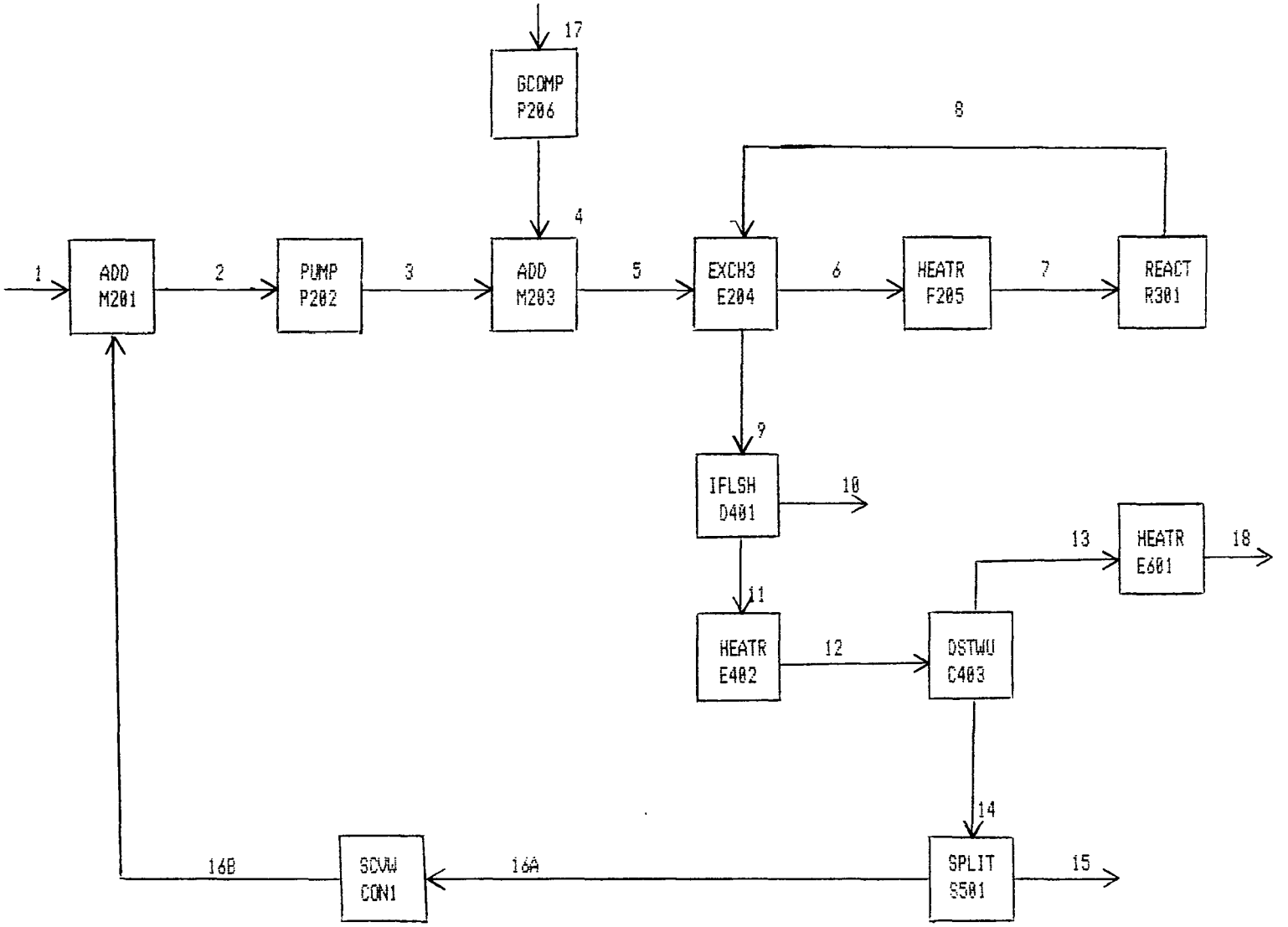


Figure 6.2 FLOWTRAN Information Flowsheet

the example section of Chapter 5, and poses the same problems. This is avoided by using the HEATR block which computes the heat required to bring a stream to a given condition with no reference to a second stream.

CHAPTER 7

CONCLUSIONS

CHESS contains both good and bad characteristics. The abilities of the program are mostly limited to simulation, but educational users do not require as extensive an array of design abilities as industry. This is due to the fact that education has the option of choosing the process, and is not forced to simulate an existing plant. The worst part of CHESS's abilities is it's limited component library. This is also considered the worst characteristic of CHESS in general.

As far as software is concerned, CHESS works very well. The program often has trouble converging, but this can be solved by giving an initial estimate of the recycle stream.

The input for the program can be improved if a preprocessor is added, but the purpose of this research is not to change the programs, but to try to get the programs to work in as close to the original condition as possible, and then to evaluate them. The output is very clear and contains a good balance between minimum pages and needed information.

The structure of CHESS is good. It is very simple to install, and executes quickly and inexpensively. This is the best characteristic of CHESS.

The PROCESS program is very impressive. It's abilities are extensive. This is probably due to it's intended use. It was not intended to serve education but to be able to simulate almost any process found in industry today. Unfortunately, the memory and money required to use the program cannot be overlooked.

For FLOWTRAN, the VAX version is the only version which should be considered in our circumstances, since it is the only version which will execute.

The best parts of FLOWTRAN are it's abilities and structure. Some abilities which are important are it's case study options and controllers over any parameter, the ability to add FORTRAN lines between unit simulations, component database, and costing. These abilities make the program very versatile. These are not as extensive as the abilities of the PROCESS package, but are better than those of the CHESS program.

The structure of the FLOWTRAN system is also good. It is very easy and inexpensive to install and execute. When running the examples which came with the FLOWTRAN program, no problems occurred.

The worst part of the FLOWTRAN package is its interface. The format for the input data is not easy to remember. This is balanced, though, by good documentation.

The recommendation of which program is best to use for educational purposes and the current computers available at the University of Arizona is the FLOWTRAN package. It contains virtually all of the abilities of the CHESS program, with the exception of a fired heater and a steam heater, but adds better controllers, the ability to add additional models and FORTRAN statements, a larger component data base, and process costing abilities.

APPENDIX A

REVISED CHESS MANUAL

The information presented here is a condensation of the CHESS User's Guide written by R. L. Motard and H. M. Lee (1). It was not meant to replace the guide, but to provide a shortened version to distribute to students. This manual also attempts to correct mistakes found in the Guide due to changes made in the CHESS program, and in the computer used. For specific techniques used in CHESS, the User's Guide should be consulted.

CHESS has the capacity to handle 100 streams, 50 equipment units, 20 components, and 7 streams per unit in any given simulation. A complete Chao-Seader physical properties package as modified by Grayson-Streed is provided for 98 compounds. More can be added if needed. The limits of the physical property package are:

1. regular liquid solutions
2. ideal vapor solutions
3. For hydrocarbons (except methane)

   Tr = 0.5 to 1.3 based on the pure component Tc

   Pressure <= 2000 psia and Pr <= 0.8 of the system
4. For light components (Hydrogen and Methane)

   Temperature = -100 F to Tr = 0.93 (or 500 F)

Pressure <= 8000 psia

Concentration <= 20 mole % of the other dissolved gases in the liquid

A-1 Preparing Input Data

The first step in using CHESS is to make a flowsheet. This flowsheet is similar to a process flowsheet except for two things: each unit can be represented by a box; and extra units must be added to represent points where streams join and separate. Using this flowsheet, the input data can be prepared. An example of a process flowsheet and a corresponding CHESS information flowsheet is given in Figures 3.1 and A-1. Note the addition of MIXR and DVDR blocks.

The input data can be divided into eight sections: 1. a problem type card; 2. a title card; 3. the process configuration and component data; 4. the equipment parameters; 5. the external stream parameters; 6. the internal stream parameters; 7. the recycle calculation options; and 8. any additional component data.

1. Problem Type The problem type card was originally meant to indicate whether or not the data was for a new problem or a variation of the last problem. This is no longer possible. The only entry for this section now is the word CLEAN entered in column 1 on the first data card.

2. Title The title card comes second. This can be any

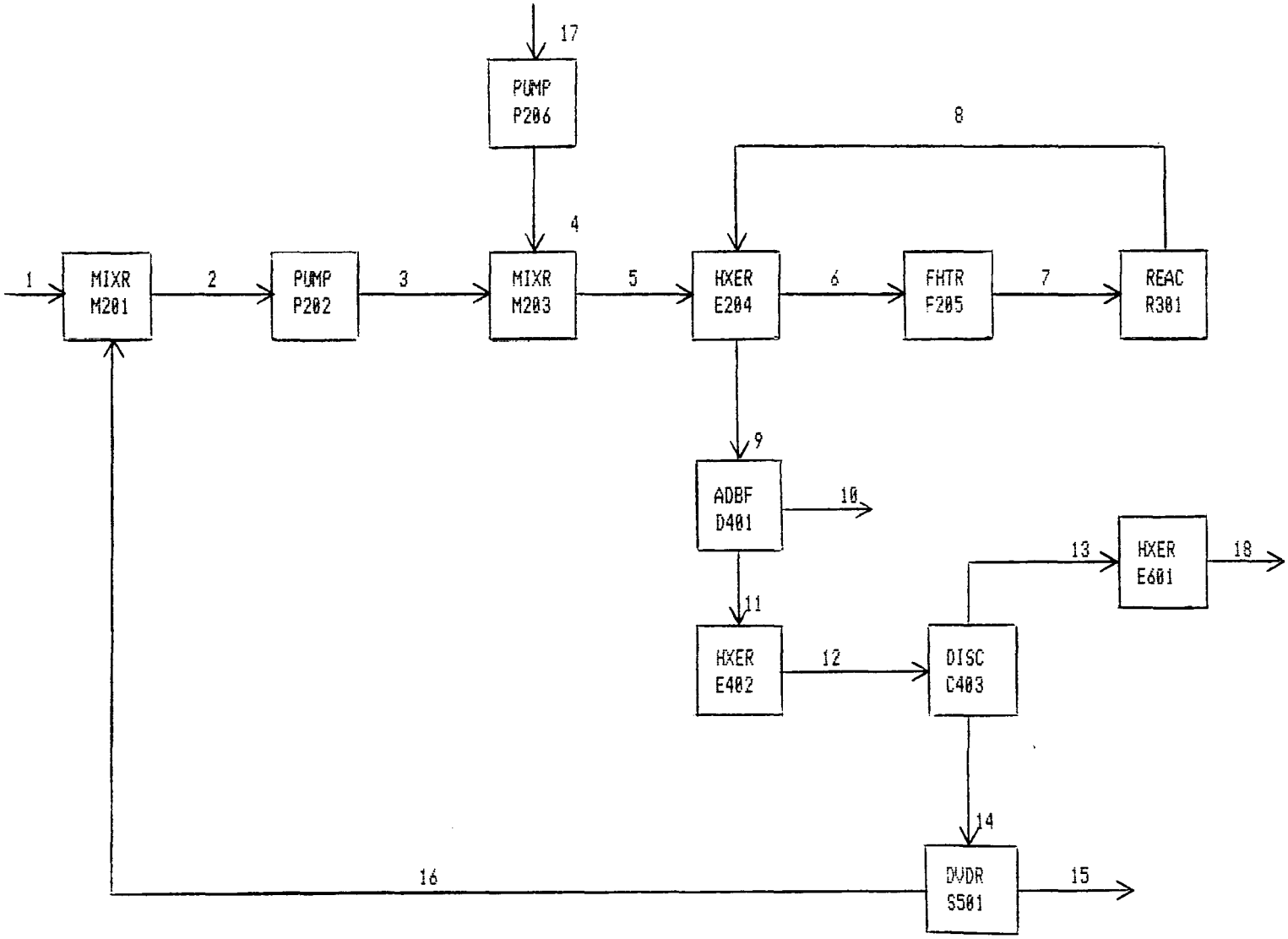


Figure A 1 CHESS Information Flowsheet

string of 79 characters, and is typed on the second card starting in the second column. The first space is left blank because it is a carriage control character for the printer.

3. Process Configuration and Component Data The process configuration data is determined from the flowsheet which was made initially. It consists of a unit number, a four character external unit name, the name of the model used to represent the equipment, and the numbers of the streams entering and exiting from the unit. The equipment numbers should all be serial. If one number is skipped, the program will not work correctly. A summary of the models available to use with CHESS are given in Table A-1. Section A-2 gives a more thorough description of each model and the input required.

This data is inserted in the input directly following the title in the following format:

1. the equipment number in columns 1 and 2;
2. the model name in columns 3 through 6;
3. the external name in columns 7 through 10;
4. the input stream numbers starting in column 11 with 3 columns allowed per stream;
5. the negative of the output stream numbers following the input streams in the same format; and
6. enough zeros to complete ten entries on the card in input stream format. If an entry does not fill the

Table A-1 Model Summary

1. ABSR - Absorber/Stripper
2. ADBF - Equilibrium Flash
3. BOTT - Bottoms or Vaporization Controller
4. CTRL - Controller for component ratio, stream temperature, or flow
5. DISC - Short-Cut Distillation Rating and Design
6. DISR - Short-Cut Distillation Rating
7. DIST - Simple Component Split Distillation
8. DVDR - Divider
9. ENTM - Entropy Machine Routine
10. FHTR - Direct Fired Heater
11. HXER - Heat Exchanger/Condenser
12. LQLQ - Liquid-Liquid Extraction
13. MIXR - Mixer with Flash
14. MMIX - Mixer
15. MSEQ - Multistage Equilibrium
16. OVHD - Overhead or Condensation Controller
17. PUMP - Pump/Compressor
18. REAC - Reactor
19. VALV - Valve/Downstream Pressure Controller
20. ADD1 through AD14 - User Added Subroutines

entire column space allocated, the entry should be right justified. This is repeated for each unit with one unit per input card. When completed, repeat the above procedure once more with zeros.

For some models, a special order must be followed for the output stream designations. If so, this is discussed in Section A-2 in the section describing that particular model. In general, always list a vapor output stream before a liquid output stream. The process configuration data for the flowsheet in Figure A-1 is given in Figure A-2.

The component data follows the process configuration data. CHESS requires the number of components to be used, NOCOMP, and the component identification numbers of these compounds, COMPNT vector. The compounds which CHESS is able to use and their respective component ID numbers are listed in Table A-2. To add a component, specify a component ID number greater than 100, and the additional component data described later. The number of components is specified first on a card in columns 1 through 3. Then, on the next card, the component identification numbers are given with three spaces per entry.

If you wish to change the name of the component, specify the KOMNAM array element number for that component in columns 1 through 2. If not, place a zero. The KOMNAM array stores the names of the components being used. There

```
 1MIXRM201   1 16 -2   0   0   0   0
 2PUMPP202   2 -3   0   0   0   0   0
 3PUMPP206  17 -4   0   0   0   0   0
 4MIXRM203   3  4 -5   0   0   0   0
 5HXERE204   5  8 -6  -9   0   0   0
 6FHTRF205   6 -7   0   0   0   0   0
 7REACR301   7 -8   0   0   0   0   0
 8ADBFD401   9-10-11   0   0   0   0
 9HXERE402  11-12   0   0   0   0   0
10DISCC403  12-13-14   0   0   0   0
11DVDRS501  14-15-16   0   0   0   0
12HXERE601  13-18   0   0   0   0   0
```

Figure A-2 Process Configuration Data For Figure A-1

Table A-2 Component ID Numbers

1.	Hydrogen	50.	Hydrogen Sulfide
2.	Methane	51.	Sulfur Dioxide
3.	Ethane	52.	2-Methyl-C5
4.	Propane	53.	3-Methyl-C5
5.	I-Butane	54.	2,2-Di-Cl-C4
6.	N-Butane	55.	2,3-Di-Cl-C4
7.	I-Pentane	56.	1-Heptene
8.	N-Pentane	57.	Propadiene
9.	Neo-Pentane	58.	1,2-Butadiene
10.	N-Hexane	59.	C2-Cyclo-C5
11.	N-Heptane	60.	C2-Cyclo-C6
12.	N-Octane	61.	Isoprene
13.	N-Nonane	62.	Water (vapor only)
14.	N-Decane	63.	Ammonia
15.	N-Undecane	64.	Carbon
16.	N-Dodecane	65.	Acetylene
17.	N-Tridecane	66.	Propyne
18.	N-Tetradecane	67.	1-Butyne
19.	N-Pentadecane	68.	2-Methylpropene
20.	N-Hexadecane	69.	Cyclopentene
21.	N-Heptadecane	70.	N-Propylbenzene
22.	Ethylene	71.	I-Propylbenzene
23.	Propylene	72.	1-Cl-2-C2-Benzene
24.	1-Butene	73.	1-Cl-3-C2-Benzene
25.	Cis-2-Butene	74.	1-Cl-4-C2-Benzene
26.	Trans-2-Butene	75.	1,2,3-Mesitylene
27.	I-Butene	76.	1,2,4-Mesitylene
28.	1,3-Butadiene	77.	1,3,5-Mesitylene
29.	1-Pentene	78.	N-Butylbenzene
30.	Cis-2-Pentene	79.	2-Methylhexane
31.	Trans-2-Pentene	80.	3-Methylhexane
32.	2-Methyl-1-Butene	81.	2-Methylheptane
33.	3-Methyl-1-Butene	82.	2,2,4-Trimethyl-C5
34.	2-Methyl-2-Butene	83.	1-Octene
35.	1-Hexene	84.	Cyclopentene
36.	Cyclopentane	85.	Tns-1,3-Di-Cl-CyC6
37.	Methylcyclopentane	86.	Cis-1,4-Di-Cl-CyC6
38.	Cyclohexane	87.	Tns-1,4-Di-Cl-CyC6
39.	Methylcyclohexane	88.	1,1-Di-Cl-CyC6
40.	Benzene	89.	Cis-1,2-Di-Cl-CyC6
41.	Toluene	90.	Cis-1,3-Di-Cl-CyC6
42.	O-Xylene	91.	N-Octadecane
43.	M-Xylene	92.	N-Nonadecane
44.	P-Xylene	93.	N-Eicosane
45.	Ethylbenzene	94.	1,1-Di-Cl-CyC5
46.	Nitrogen	95.	Cis-1,2-Di-Cl-CyC5
47.	Oxygen	96.	Tns-1,2-Di-Cl-CyC5
48.	Carbon Monoxide	97.	Cis-1,3-Di-Cl-CyC5
49.	Carbon Dioxide	98.	Tns-1,3-Di-Cl-CyC5

are four entries in the array for each component. The KOMNAM array element number for the Ith component in the component data would then be 4(I-1)+1. For example, to change the name of the fourth component listed in the input data, enter 4(4-1)+1 or 13 in columns 1 through 2. The new name goes on the next card starting in column 2. The maximum length of the name is 15 characters. This is repeated until a zero appears as the KOMNAM array element.

For example, if a system was using octane, nonane, decane, undecane, propyne, and i-butane, the data would appear like this:

```
 6
 12 13 14 15 66  5
 0
```

If the name of i-butane was to be changed to simply butane, the data would be

```
 6
 12 13 14 15 66  5
21
 BUTANE
 0
```

The KOMNAM array index was 21 because i-butane was the sixth component listed in the input data.

The last entry in the component data section is a zero in column 2. This is the variable IPUNCH. All of the

options available for this variable were deleted before the program arrived here.

4. Equipment Parameters Unlike the process configuration and component data, the equipment parameter data can be in any order provided the strings $EQLIST and $END appear in column 2 at the beginning and end, respectively, of the data. The format for the data in this and the rest of the sections is

variable = value,

and

array name= value 1, value 2, etc.,

starting in column 2, and ending with a comma. This is known as FORTRAN NAMELIST format. The number of values given for the arrays must equal the size of the array. Multiple entries of the same value can be entered using I*X, where X is the value and I is the number of consecutive elements given that value. If all of the variables in an array are to be zero, do not input anything. All variables and arrays not specified will default to zero unless mentioned otherwise. An error will result if the entry is not followed by a comma, or started in column 1.

The arrays which need to be defined in the equipment data section are ENAME and EQPJ. ENAME is an integer array used to specify the equipment numbers which are being used in the simulation but do not have an EQPJ array given. The size of this array is 50.

EQPJ is a real, equipment parameter array which must be specified for each unit not listed in ENAME. The size of the array is 25. To specify which unit the vector corresponds to, replace the letter J in EQPJ with the unit number. Each EQPJ vector should be given on a separate card. The parameters which are inputted in the array vary from unit to unit, and therefore, are mentioned in Section A-2 under the appropriate section.

5. Extensive Stream Parameters The extensive stream parameter data is next. It is used to define the composition and enthalpy of streams. The beginning and end of this section of data is marked by the strings $SEXLST and $END starting in column 2.

For each input stream, a SEXJ vector must be given as described in Figure A-3. All entries should be real. The size of the array is 23. A vector can also be given as an initial estimate for recycle streams to improve convergence. Note the minimum required variables are listed at the bottom of Figure A-3. As long as these are given, the state of the stream is completely defined, and the rest of the variables can be calculated by the program. The letter J is replaced by the stream number in SEXJ.

The integer array SNAME must also be given to list the streams being used in the simulation but not described in SEXJ vectors. The size of this array is 100.

SEXJ = J, h, TM, C_1, ..., C_n, M*0.,

where

J = Stream number

h = Heat Content, BTU/lb mole

TM = Total moles per hour

C_i = Stream composition in lb. moles/hr of component i

N = Total number of components used, NOCOMP

M = Balance of vector, up to 23 elements, is filled out with zeros

Minimum required variables:

J, TM, and C_1, ..., C_n

Figure A-3 Extensive Stream Parameters

SINJ = J, F, 0., VF, T, P, H, Visc, k, Z_l, Z_v,

where

F = Stream flag; 0. = intermediate stream

1. = feed stream

2. = product stream

VF = Vapor fraction, moles vapor/total moles

T = Temperature, R

P = Pressure, psia, default 0.01

H = Total heat content, BTU/hr

Visc = Viscosity

k = Thermal Conductivity

Z_l = Compressibility factor for the liquid phase

Z_v = Compressibility factor for the vapor phase

Minimum required variables:

Input streams: J, F, P, and T or VF

Output streams: J and F

Figure A-4 Intensive Stream Parameters

6. Intensive Stream Parameters The intensive stream parameter data follows the extensive stream data, and is bounded by the strings $SINLST and $END. This section also includes the array SNAME except now SNAME refers to all of the streams not mentioned in SINJ vectors and used in the simulation.

A SINJ vector must be given for each input and output stream. Figure A-4 gives the variables which can be specified as well as the minimum requirements. There can only be ten entries for the SINJ vector even though eleven are listed. The zero for the third element was a mistake in the program which made the specification of Z_v impossible, but not the output.

7. Recycle Calculation Options The recycle calculation data is optional. This is because a decomposition method proposed by Barkley and Motard is included in CHESS. If recycle information is to be specified, any one or set of the variables LOOPS, NPFREQ, DERROR, and KTRACE, or of the arrays KE2, KE3, and KE4 should be given.

LOOPS is the maximum number of iteration loops allowed in the recycle calculations. It has a default value of 20. The program will print an error message should this many iterations be tried without convergence, and then continue as if the calculations had converged.

DERROR is the absolute maximum error between recycle iterations to define convergence. The default value is 0.01.

NPFREQ is an integer number specifying the iteration interval at which an intermediate stream matrix summary is printed.

KTRACE also controls output printing. If KTRACE equals 0, no tracing of the calculations is done. If KTRACE equals 1, a tracing of the equipment modules called will be made together with a summary of the convergence promotion procedure. KTRACE equals 2 is the maximum tracing output. It prints all of the information for KTRACE equals 1, in addition to, a summary of the input streams for the module being calculated.

If a special order is wanted in the equipment calculations, a list of equipment numbers should be given in the KE2 integer array. Each unit can be listed more than once, and units not in the recycle loop should be omitted. The size of this array is 50.

The KE3 array lists stream numbers to be used in the recycle net decomposition routine (the tear streams). The KE4 array also lists stream numbers, except it lists the streams which will be used for convergence acceleration. Both of these arrays are dimensioned for 10. Normally only KE2 is given and then KE3 and KE4 are calculated by the computer.

8. Additional Component Data The final section of input is the additional component data. This data is needed if a component identification number greater than 100 was given in the process configuration and component data section. This section is indicated by the strings $NSCOMP and $END appearing at the beginning and end, respectively.

The required data is shown in Table A-3. Each of these vectors is dimensioned for 20 entries.

For petroleum fractions, specify a component ID number greater than 200, required data numbered 14 through 17, and then the rest of the data will be calculated by CHESS as described in the CHESS User's Manual on pages 49 - 51.

This is all of the input data required except when LQLQ is used. Table A-4 summarizes all of the possible variables to be input and the order they should be given. Section A-7 contains the complete input data and output results for the process shown in Figure A-1.

A-2 Model Information

This section describes the models available and gives the equipment parameters required for their use in CHESS. In all of the equipment parameter vectors, J will always refer to the equipment number given to the unit in the process configuration data and is a real variable. The term "stream number" will refer to the order the stream is

Table A-3 Component Data Input

1. APC(I) - critical pressure, psia
2. ATC(I) - critical temperature, R
3. AVC(I) - critical volume, cm^3/g mole
4. AMW(I) - molecular weight
5. AOMEG(I) - accentric factor
6. ADEL(I) - solubility parameter, $(cal/cm^3)^{1/2}$
7. AVW(I) - characteristic molar volume, cm^3/ g mole
8. APH(I) - a term in $Cp^o = a+bT+cT^2+dT^3+eT^4+fT^5$
9. BET(I) - b term in Cp^o
10. GAM(I) - c term in Cp^o
11. DTA(I) - d term in Cp^o
12. EPS(I) - e term in Cp^o
13. ETA(I) - f term in Cp^o
14. BPM(I) - mean average boiling point
15. BPMOL(I) - molal average boiling point
16. BPCA(I) - cubic average boiling point
17. API(I) - API gravity

where

Cp^o = zero pressure heat capacity in cal/mole K

I = position of the non-standard component in the COMPNT vector.

Table A-4 Input Data Summary

1. Problem Type - CLEAN in column 1

2. Title - 79 characters starting in column 2

3. Process Configuration and Component Data -

For each equipment: Number (columns 1 & 2), model name (columns 3 - 6), external name (columns 7 - 10), stream configuration (starting in column 11, with 3 columns per entry)

Number of components (NOCOMP) in columns 1 & 2

Component ID numbers used (COMPNT vector) 3 columns per entry

KOMNAM index or 0 in columns 1 & 2

new name if above not zero

IPUNCH - 0 in column 1

4. Equipment Parameters - EQPJ(25), and ENAME(50)

5. External Stream Parameters - SEXJ(23), and SNAME(100)

6. Internal Stream Parameters - SINJ(10), and SNAME(100)

7. Recycle Calculation Options - LOOPS, NPFREQ, DERROR, KTRACE, KE2(50), KE3(10), and KE4(10)

8. Additional Component Data - APC(10), ATC(10), AVC(10), AMW(10) ,AOMEG(10) ,ADEL(10) ,AVW(10), APH(10), BET(10), GAM(10), DTA(10), EPS(10), ETA(10), BPM(10), BPMOL(10), BPCA(10), and API(10)

given in the input configuration (Not the actual stream number shown on the flowsheet). The same is true for the component number. It is the order number not the ID number. For more detailed descriptions of the calculational methods, see the User's Manual.

1. ABSR - Absorber/Stripper

This routine is used to model the multicomponent separation of a vapor-liquid system. The order the stream input should be in is input gas, input oil, output gas, and output oil. The output oil temperature can be calculated in two ways. If the variable mode equals zero, the temperature will be computed by an enthalpy balance. If mode equals one, the exit oil temperature will be the bubble point temperature.

The exit gas temperature is the dew point of the stream provided this is within a limit, DT, from the entering oil temperature. If the dew point temperature falls outside of this limit, the temperature will be the average of the dew point and the nearest limit. The default value for the limit is $\pm$ 10 F, but this can be set to a larger value as well.

The equipment parameter vector is

EQPJ = J, N, DT, mode, 21*0.,

where

N = the number of equilibrium stages,

DT = the temperature limit discussed earlier, and

mode = the exit oil temperature calculation method

0. for enthalpy balance

1. for bubble point temperature.

2. ADBF - Equilibrium Flash

This module is used to either isothermally or adiabatically flash a stream. An isothermal flash in simulation is not exactly the same as an isothermal flash in thermodynamics. In simulation, an isothermal flash is one in which the final temperature is specified. The change in temperature does not have to be zero.

In the stream configuration data, either one or two output streams may be specified. If two are specified, the first output stream will be the vapor and the second will be the liquid. If one output is given, the stream will have a mixed phase.

The equipment parameter vector is

EQPJ = J, mode, EQP(3), EQP(4), 21*0.,

mode = 0., adiabatic equilibrium flash with prior bubble point and dew point calculation using the input pressure and enthalpy (Input temperature and vapor fraction are updated),

mode = 1., constant temperature equilibrium flash using input temperature and pressure (Input enthalpy and vapor fraction are updated),

mode = 2., similar to mode = 0. except no bubble point

and dew point calculations will be done first,

mode = 3., same as mode 0. or 2. with retained V and K values,

mode = 4., a flash is made to a specified vapor fraction and pressure in EQP(3) and EQP(4),

mode = 5., isothermal flash to a temperature and pressure specified in EQP(3) and EQP(4), respectively,

mode = 6., adiabatic flash to a temperature specified in EQP(3) by adjusting output pressure,

mode = 7., isothermal flash to vapor fraction and temperature specified in EQP(3) and EQP(4) by adjusting output pressure,

mode = 8., adiabatic flash to output pressure specified in EQP(3),

mode = 9., flash to a fractional recovery in the vapor phase specified in EQP(3) of a component specified in EQP(4) by adjusting output temperature, and keeping pressure constant, and

mode = 10., same as mode 9 except temperature is kept constant and pressure is varied.

If EQP(3) or EQP(4) is left blank when needed, the input values will be used.

3. BOTT - Bottoms or Vaporization Controller

This model is used to simulate a reboiler for the MSEQ unit, or a vaporizer. For a partial reboiler, the first output stream, V, listed in the process configuration

data should be the vapor stream, and the second output stream, B, should be the liquid. For a total reboiler both, of course, will be liquid.

The equipment parameter vector is

EQPJ = J, mode, specification, 22*0.,

where

mode = 1., total reboiler with reflux ratio specified, V/B,

mode = 2., total reboiler with reflux amount specified, V,

mode = 3., partial reboiler with reflux ratio specified, V/B,

mode = 4., partial reboiler with reflux amount specified, V,

mode = 5., total reboiler with bottoms amount specified, B,

mode = 6., partial reboiler with bottoms amount specified, B, and

specification = value for reflux ratio, reflux, or bottoms amount.

If a total amount for a stream is given which is greater than the input stream amount, all of the input will go to that particular stream.

4. CTRL - Controller

This routine is used to control component ratio, stream temperature, or flow rate for the ABSR, ADBF, MIXR, or DISC routines by adjusting the input stream pressure, the input stream temperature, or an input or make-up stream flow rate. The controller model replaces the model being controlled. For example, to control an absorber/ stripper, do not use the ABSR module but the CTRL module in the ABSR mode.

The equipment parameter vector is

EQPJ = J, P1, ..., P13, COB, mode, CRT, NSI, NSO, CA, CB, To1, XMAX, XMIN, 0.,

where

P1, ..., P13 = normal equipment parameters for the model being controlled (See EQPJ for that unit),

COB = control objective:

1., component ratio control, CA/ CB,

2., temperature control,

3., flow rate control,

mode = 1., adjusting temperature,

2., adjusting pressure,

3., adjusting flow rate,

add to this: 00 for controlling a MIXR model,

10. for an ABSR model,

20. for a DISC model,

30. for an ADBF model,

CRT = desired value for COB in (lb moles/hr)/(lb moles/hr), R, or lb moles/hr depending on COB,

NSI = input stream number which will be adjusted,

NSO = output stream number which will be controlled,

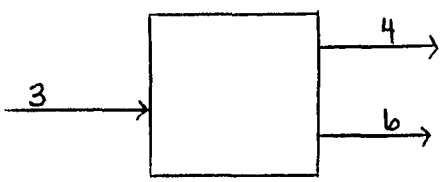
CA = number of the component in the numerator,

CB = number of the component in the denominator,

Tol = absolute tolerance of CRT,

XMAX = upper limit for the controlling variable, and

XMIN = lower limit for the controlling variable.



Consider the reboiler shown above which would normally be modeled by ADBF, and whose input temperature in stream 3 will be used to force the output composition ratio of CH_4/C_2H_6 in stream 6 to be 0.02 ± 0.001.

The first step in determining the equipment parameters is to determine what the parameters for the model would be without the controller. For ADBF, this is mode = 1 (flash at input temperature and pressure to determine H and V/F). The variables P1 through P13 are these parameters without the J entry i.e. 1., 12*0.

The control objective, COB, is 1 for component ratio. To determine the mode, decide how the objective should be achieved and add to it an offset for the equipment being controlled. Adjusting temperature is 1., and controlling ADBF is 30. The mode is then 30 + 1 or 31. CRT is 0.02, and Tol is 0.001.

To determine NSI and NSO, refer to the process configuration data for that unit, which is

1CTRLCON1 3 -4 -6 0 0 0 0

for this example. The temperature will be varied in stream 3. This is the first stream in the input stream list, so NSI equals 1. The composition ratio will be forced in stream 6. This is the second stream in the output stream list, so NSO equals 2. NSI and NSO are <u>not</u> the stream numbers given on the flowsheet.

The following was the order the components were given in earlier data

49 46 2 3 4 5 6 7 8 11

Note CH_4 is component number 3 on the list and C_2H_6 is number 4. This means CA = 3, and CB = 4. CA and CB are not the component ID numbers. The range the temperature can be varied is from 600 to 620 R which makes XMAX 620 R and XMIN 600 R. The EQPJ vector is then

EQP1 = 1.,1.,12*0.,1.,31.,0.02,1.,2.,3.,4.,0.001,620.,600.,0.,

Obviously, special care must be taken to prepare the input data for this model.

The process configuration data will be the same as the unit being controlled. Therefore, refer to that section for any special instructions.

5. DISC - Shortcut Distillation Rating and Design

This routine models distillation units by using correlations such as Fenske's total reflux equation, Underwood's equations for minimum reflux, Gilliland's correlation for stage and reflux ratio, etc. The actual correlation used depends on the option. Four different options are provided. Three are for design, and one is for rating.

The equipment parameter vector is

EQPJ = J,option, LK, FLK, HK, FHK, cond, R, R/RM, S, 15*0.,

where

option = 1., rating using Fenske-Underwood stage and reflux equations, and Gilliland feed tray equations,

=2., design, same as option 1 except Fenske feed tray location correlation is used,

=3., design, same as option 1 except Kirkbride feed tray location correlation is used,

=4., design, Henstebeck's equivalent binary approach is used to establish an x-y relationship after

which the McCabe-Thiele method is used for stage and feed tray calculations,

LK = light key component number,

FLK = fraction of light key to distillate (specify for design option only),

HK = heavy key component number,

FHK = fraction of HK to distillate,

cond = indication of type of condenser,

0. = total condenser,

1. = partial condenser,

R = reflux ratio (specify for rating option only),

RM = minimum reflux ratio (specify R/RM for design option only), and

S = theoretical number of stages (specify for rating option only).

If using a partial condenser, only a vapor product will be withdrawn. When specifying the input and output stream configuration, always list the distillate stream first.

6. DISR - Shortcut Distillation Rating only

This model is exactly the same as the DISC model except the design options have been deleted. The equipment parameters, therefore, are identical to those given for DISC when option 1 is chosen, i.e.

EQPJ = J, 1., LK, 0., HK, FHK, cond, R, 0., S, 15*0.,

Refer to the previous section for definitions of the terms.

7. DIST - Simple Distillation

This model will split one stream into two streams on a component by component basis using split fractions given as input parameters. The conditions of the streams can then be determined in various ways depending on the mode.

The equipment parameter vector is

EQPJ = J, mode, 0., F_1, ..., F_m, N * 0.,

where

mode = 0., an isothermal split assuming top as vapor and bottom as liquid,

= 1., the top stream is set to its dew point, and the bottom stream to its bubble point,

= 2., both output streams are set to their dew point,

= 3., both output streams are set at their bubble point,

= 4., K-values are given for the stream components instead of F_i,

F_i = fraction of the entering component i found in the overhead output stream, and

N = the number of zeros needed to complete 25 elements.

As with the other distillation models, always list the distillate stream first in the output stream configuration.

8. DVDR - Divider

This routine splits one input stream into up to six different output streams. This is different from DIST because each stream will have the same composition.

The equipment parameter vector is

EQPJ = J, F_1, ..., F_m, N*0.,

where

F_i = fraction of the input stream which goes into the ith stream listed in the process configuration data for this block, and

N = the number of zeros needed to complete 25 elements.

9. ENTM - Entropy Machine

This routine is intended to calculate the entropy change of a stream due to a variation in pressure. The information needed is simply the output pressure and the efficiency. The routine will then calculate the actual and theoretical heat lost, and the entropy of the feed. Ideally the stream should have a single phase.

The equipment parameter vector is

EQPJ = J, 0., POUT, EFF, 21*0.,

where

POUT = output pressure, and

EFF = efficiency, default = 0.8.

10. FHTR - Direct Fired Heater

This routine calculates the fuel gas consumption required to heat a stream to a desired outlet temperature. To enable a rating evaluation to be done, the specified maximum heat duty of the heater must be inputted. If the heat required to raise the stream to the given temperature is more than that amount, the outlet temperature will be set to the maximum possible for that rating. The heating medium is fuel gas with a heat content of 900 BTU/SCF.

One or two output streams may be specified. If two are specified, the first will be assumed to be the vapor stream and the second the liquid.

The equipment parameter vector is

EQPJ = J, QR, DP, T, 21*0.,

where

QR = rated heat duty, MBTU/hr,

DP = pressure drop in the unit, psia, and

T = desired outlet temperature, R.

11. HXER - Heat Exchanger/ Condenser

This model is used to represent either a counter-current heat exchanger, or a stream being cooled using cooling water. The water is at 550 R and will rise 15 R in the heat exchanger. This can be changed. If the output temperature specified for a single stream heat exchanger is less than 550 R, or the stream needs to gain heat from the water, the model will still compute the heat duty but not

the area of the exchanger. Modes 1 and 4 will rate a given heat exchanger, and modes 2, 3, 5, and 6 will size an exchanger for a given output condition.

The equipment parameter vector for HXER is

EQPJ = J, U, AR, B, A, C, mode, DT or T, DP_1, DP_2, Q, 14*0.,

where

U = overall heat transfer coefficient, BTU/(hr ft^2 R),

AR = area per shell, ft^2/shell,

B = number of shells in series,

A = number of shell passes/shell,

C = number of tube passes/shell,

mode = type of heat exchanger and specifications required,

For two input streams:

mode = 1., counter-current heat exchanger with U, AR, B, A, C, mode, DP_1, and DP_2 specified;

mode = 2., stream one will be condensed while the temperature is held constant and then cooled. Specify U, B, A, C, mode, DT, DP_1, and DP_2 (If the first stream is not a vapor, mode 3 will be used);

mode = 3., counter-current exchanger with a minimum temperature approach given. Specify U, B, A, C, mode, DT, DP_1, and DP_2.

For one input stream:

mode = 4., standard water cooled exchange using cooling water. The heat exchanger geometry and heat transfer coefficient are known. Specify U, AR, B, A, C, and mode.

mode = 5., simple temperature controller. Given the output temperature, the heat duty is calculated. Nothing about the exchanger geometry or heat transfer coefficient needs to be known. Specify T and mode.

mode = 6., same as mode 4 except the process output temperature is known and AR is calculated.

If the mode is greater than 6, the mode will be assumed to be the cooling water input temperature, and T the cooling water output temperature. Mode 4 will then be used.

DT or T = minimum approach temperature difference (DT), or desired output temperature (T) in R depending on the mode chosen, the default is 20 R,

DP_1 = pressure drop for the first input stream, psia,

DP_2 = pressure drop for the second input stream, psia, and

Q = heat duty for stream 1, MBTU/hr.

To preserve continuity, the first input stream will correspond to the first output stream, etc. in the process configuration data.

12. LQLQ - Liquid-Liquid Extractor

This routine is used to model a liquid-liquid extractor. Unlike the other modules, this module is limited to six components, and requires more input than just an EQPJ vector. The additional input is shown in Table A-5. This input follows all of the other input data described in Section A-1. The format is F8.4, which means that only four digits past the decimal place can be given, the number cannot be larger than 999.9999, and each entry is allotted 8 columns on a card. A maximum of 9 entries can be given on a card, and each different array starts on a new card.

Four different feed streams and two output streams are allowed. All streams can enter on or exit from any stage.

For each feed stream, a design specification can be made on any stage in the column. The maximum number of specifications is two. These specifications can be the raffinate rate, extract rate, raffinate composition, or extract composition. The specifications are achieved by controlling the feed streams.

The EQPJ vector is:

EQPJ = J,NSTAGE, FD1, FD2, FD3, FD4, MAX, NPRINT, EXRR, SSTAG1, SSPEC1, SSTAG2, SSPEC2, A1, B1,C1, D1, E1, A2, B2, C2, D2, E2, RMAX,

where

NSTAGE = number of stages,

Table A-5 Additional LQLQ Data

XL - Total raffinate flow rate for each stage

XV - Total extract flowrate for each stage

X(J,1) - Rafinate composition in lb moles on stage 1 of component J. Repeat for each component.

X(J,NSTAGE) - raffinate composition in lb moles on stage NSTAGE of component J. Repeat for each component.

Y(J,1) - extract composition on stage 1 of component J. Repeat for each component.

Y(J,NSTAGE) - extract composition on last stage (NSTAGE) of component J. Repeat for each component.

AA(1,J) - ternary interaction coefficient of component J and component 1. Repeat for each component.

DD(J) - Coefficient in Margules four suffix equation for component J. Repeat for each component.

CC(J) - Coefficient in Margules four suffix equation for component J. Repeat for each component.

FD_i = the stage feed stream i is fed onto,

MAX = maximum number of iterations attempted to achieve the specifications before continuing with the other units,

NPRINT = output printing code,

= 1., the input plus the converged case is printed,

= 2., a summary of all the iterations is included with NPRINT 1,

= 3., the compositions and rates for the streams is included with NPRINT 2,

= 4., all of the errors found between iterations, and the corrections made for the next iteration is added to NPRINT 3,

EXRR = extract reflux ratio,

SSTAGi = flow rate of the output stream number i (positive for rafinate, negative for extract),

Ai = the stage the specification i is on,

Bi = the type of specification for number i

= 1., raffinate rate,

= 2., extract rate,

= 3., raffinate composition,

= 4., extract composition,

Ci = component that the specification i is on if Bi = 3 or 4,

Di = value for specification i,

Ei = feed stream which will be controller to achieve specification i, and

RMAX = relative maximum step size the change in flow can undergo between iterations, default = 0.5.

13. MIXR - Mixer with Flash

This module will mix up to six input streams to form one mixed phase stream or one vapor and one liquid stream. If two output streams are specified, the first will be the vapor and the second the liquid. The pressure of the output will be the lowest pressure of any of the input streams. The temperature and vapor fraction of the output stream is determined by performing an adiabatic flash calculation.

The EQPJ vector has no entries except for J, the equipment number.

EQPJ = J, 24*0.,

If desired, the EQPj vector can be omitted, and the equipment number added to ENAME instead.

14. MMIX - Mixer without a Flash

This module also mixes up to six input streams, except only one output stream can now be given. The pressure of the output stream will be the lowest pressure of any of the input streams, and the temperature will be the molar average temperature. The vapor fraction of the output stream depends on whether more moles in the input streams were vapor or liquid. If more moles were liquid, the vapor fraction is set to 0. If not, the stream will be assumed to be all vapor.

The EQPJ vector is again empty except for the equipment number, and can be forgotten if J is added to ENAME instead.

EQPJ = J, 24*0.,

15. MSEQ - Multistage Equilibrium

This routine can simulate any process which can be treated as a series of adiabatic flash calculations. There cannot be more than five MSEQ units and a total of 80 MSEQ stages in any simulation. Neither a reboiler nor a condenser is included. The OVHD and BOTT routines are suggested for modeling the condenser and reboiler, respectively.

Two outputs are provided, a top vapor and a bottoms liquid. Up to five inputs can be specified. Two are the top reflux and the bottom vapor recycles. The input streams can enter on any stage. As in the DISC model, the first output stream should be the overhead vapor, and the second output stream should be the output bottom liquid.

The EQPJ vector is

EQPJ = J, N, S_1, ..., S_5, 0., 0., mode, 15*0.,

where

N = number of equilibrium stages, and

S_i = the input stage number for input stream number i.

If the mode is greater than zero, MSEQ will print intermediate stage results.

16. OVHD - Overhead Controller or Controlled Condensation Divider

This routine may be used as either a control block for the MSEQ overhead or as a controlled condensation divider anywhere in a system. The first output stream, D, will be vapor and the second output stream, R, will be liquid for a partial condenser.

EQPJ = J, mode, mode specification, 22*0.,

where

mode = 1., total condenser with reflux ratio specified, R/D,

mode = 2., total condenser with reflux amount specified, R,

mode = 3., partial condenser with reflux ratio specified, R/D,

mode = 4., partial condenser with reflux amount specified, R,

mode = 5., total condenser with distillate amount specified, D,

mode = 6., partial condenser with distillate amount specified, D, and

mode specification = reflux ratio, reflux, or distillate amount.

Do not try to use a total condenser with hydrogen present in the system.

17. PUMP - Pump/ Compressor

This routine will compute the work necessary to pump a liquid or compress a gas to a desired pressure. The maximum work capacity of the unit must be inputted to make sure that this work does not exceed the capacity. If it does, the output pressure will be adjusted to match the maximum capacity.

Either steam, electricity, or fuel gas can be used as a motive force. If steam is used, the enthalpy of the steam as it leaves and enters the pump is required. If several stages are specified for the compressor, perfect interstage cooling is assumed.

The equipment parameter vector is

EQPJ = J, N, W, P, D, H, 19*0.,

where

N = number of compressor stages for vapor input only,

W = work capacity of the machine, MBTU/hr,

P = desired output pressure in psia,

D = motive force (If using steam, this is the enthalpy of the entering steam, if using electricity, D = 0., and if using fuel gas, D = -1.) and,

H = enthalpy of the exiting steam.

18. REAC - Reactor

This routine will simulate a reactor given the stoichiometric coefficients for the reaction, and the conversion of a key reactant. The stoichiometric numbers

should be positive for products, negative for reactants, and zero for inert compounds. A maximum of seven compounds can be used.

The equipment vector is

EQPJ = J, C, K, F_1, ..., F_7, DHR, 14*0.,

where

C = fractional conversion of key component, K,

K = number of key component in component list,

F_i = stoichiometric numbers relative to key component, and

DHR = heat of reaction per mole of key component, BTU/lb mole.

19. VALV - Valve/Pressure Controller

This routine will recompute the temperature and vapor fraction for a stream given a new pressure, and assuming no enthalpy change. Two output streams can be specified to separate a mixed phase stream. In this case, the first output stream listed will be the vapor stream.

The equipment parameter vector is

EQPJ = J, P, 23*0.,

where

P = downstream pressure, psia.

20. ADD1 through AD14 - User Added Subroutines

These are reserved for user added models and will be discussed in the A-6 Adding Models section.

A-3 Output

If a successful simulation has been completed, the output will contain the following pieces of information: 1. an echo of CLEAN, the title, and an indication of which data sets were read; 2. CHESS simulation title page; 3. Tables of process vectors, stream connections, and other system variables; 4. Input stream data; 5. initial equipment summary with equipment list and individual details; 6. calculational history; 7. final stream results; 8. final equipment summary.

If the simulation results are not as expected, be sure to check the initial stream data, equipment summary, and stream connections carefully. The calculational history is also important to check every time to assure that all of the units have converged. See Section A-7 for a sample output, and Section A-8 for an explanation of some of the more common errors.

A-4 How To Access CHESS

The CHESS program is precompiled and stored on the CYBER. To access the program, the job control cards shown in Table A-5 should be given. If the messages "FILE NOT CATALOGUED, SN=SYSTEM" and "PF ABORT" appear in the dayfile, check the spelling of CHESS and SLD on your cards. If they are spelled correctly and the message persists, notify the instructor. The program has either been erased or not

prepared for use correctly. In either case, see Section A-5.

To run the program from the DEC, make a file containing your data, and another with the commands in Table A-6, and then send both of these files to the CYBER using the TOCDC command. The format for the TOCDC command is:

TOCDC $filename_1$,$filename_2$,etc./dayfile

The output will be printed on the CYBER printer as if cards had been used. The dayfile will appear in the directory under the name of the first file listed in the TOCDC command with the suffix .DAY. The term card and line in preparing the input data and in executing the program are interchangeable.

To run the program from the CYBER, separate the commands from the data with a 7/8/9 card, and use the card reader.

A-5 Preparing CHESS For Use

Once the modifications to CHESS were completed, a copy was made on tape 5069D using BACKUP. BACKUP is a DEC utility program used to copy disk files to and from magnetic tape. The commands to copy the files from the tape to a DEC account are given in Table A-7. The period and slash in column one are the system prompts and should not be typed in again.

Table A-6 Commands To Execute CHESS

```
Jobcard
PW,xxx.
ATTACH,CHESS,ID=SLD.
CHESS.
```

Table A-7 Commands To Unload CHESS From Tape

```
.MOUNT M9:MAGTAP/REELID:5069D
.R BACKUP
/TAPE MAGTAP
/INTERCHANGE
/RESTORE
/REWIND MAGTAP
/CHECK
/EXIT
.DISMOUNT MAGTAP
```

The MOUNT command requests that tape number 5069D be mounted on a nine track tape drive. Since the RING switch and password are not given, the operating system will not allow the tape to be written on. The is the default. Although the system prompt of a period will appear after the MOUNT command, this does not mean that the tape has been mounted. The message [OPR: MAGTAP (MTAx) MOUNTED will appear when the tape is ready to use. The letter x is the number of the tape drive being used.

Once inside the BACKUP program, the TAPE command is used to specify what the logical name of the desired tape is. This is only important if more than one tape drive is being used. The name MAGTAP is used here since it was given on the mount command.

When the BACKUP program copied the program onto the the tape, the DEC directory number was also written. The INTERCHANGE command tells the computer to ignore this number. Normally, the program would only restore the files to the area from which they were saved.

The RESTORE command causes all of the files listed after it to be copied onto the disk from the tape. Since no files are listed, the program will copy all of the files. The CHECK command is used to verify that the files on the tape are identical to those on the disk.

For more information concerning the BACKUP program, the University of Arizona computer center has manuals, as

well as a help file which is available while logged on to the DEC.

The CHESS program consists of six files. These files are listed in Table A-8 and should appear in the directory once they are restored from tape. The program is divided into separate files to allow editing at a lower cost and with less risk. All of these files in addition to a file containing the control cards for the CYBER need to be sent to the CYBER to compile and store the program.

The control cards for the CYBER are best handled by creating a file with the DEC editor, SOS, containing only them, and sending this file to the CYBER along with the program. The commands needed are listed in Table A-9. The letter z is the number of days the file should stay in storage. For illustrative purposes this file is called BN.CDC. To send this file and the program files to the CYBER, type

```
TOCDC BN.CDC,0V00/NOEOR,0V10/NOEOR,0V20/NOEOR,0V30/NOEOR,
    0V40/NOEOR,0V50/NOEOR,0V5K
```

with out touching the return key until the end. The /NOEOR is needed to prevent the DEC from adding an end of file mark between files.

When this is completed successfully, the dayfile will look similar to that in Figure A-5. The letter 'a' represents the time the CYBER required to compile the

Table A-8 Files Containing The CHESS Program

OV00 - main overlay

OV10 - primary overlay number 1

OV20 - primary overlay number 2

OV30 - primary overlay number 3

OV40 - primary overlay number 4

OV50 - primary overlay number 5

OV5K - secondary overlays for primary overlay number 5

Table A-9 Commands To Compile And Store CHESS

```
Jobcard.
PW,xxx.
REQUEST,CHESS,PF.
FTN,L=0.
LOAD,LGO.
NOGO,CHESS.
CATALOG,CHESS,CHESS,ID=SLD,RP=z.
```

Figure A-5 Dayfile For Compilation And Storage

```
Jobcard.
PW,xxx.
REQUEST,CHESS,PF.
FTN,L=0.
    a   CP SECONDS COMPILATION TIME
LOAD,LGO.
NOGO,CHESS.
CATALOG,CHESS,CHESS,ID=aaa,RP=z.
INITIAL CATALOG
    b PF BLOCKS HELD,  LIMIT=c.  (may not give limit)
CT  ID=SLD  PFN=CHESS
etc.
```

program, and the letters 'b' and 'c' represents the number of storage blocks used by the program and the maximum possible which could have been used by the program, respectively.

A typical student account on the CYBER cannot be used to store CHESS unless special provision has been made. The special provision is 1000 blocks of permanent file storage.

A-6 Adding Models

CHESS has the ability to have fourteen models added. These are named ADD1 through AD14. Information is transferred to and from models through COMMON blocks. These are given in Table A-10. Table A-11 defines each of the terms. A process module can refer to any of the flash routines or thermodynamic routines, but not to another unit.

The thermodynamic properties available are listed in Table A-12. To obtain any of these properties, treat them as subroutines. ARG is the order number for the stream, positive for input streams, and negative for output streams, ANS is the variable the value will be assigned to, and LIST is the vector containing the set of k-values computed for the stream.

For example, if the enthalpy of the first output stream listed in the process configuration data is wanted,

Table A-10 COMMON Block Storage

```
COMMON/EQPA/EQPAR(25,50),NEMAX,MAXEQP
COMMON/CONTL/NIN,NOUT,NOCOMP,NE,NEN,DERROR,LOOP,LOOPS,KTRACE
COMMON/STRMIN/SINUM(8),SIFLAG(8),SITYPE(8),SIVPFR(8),SITEMP(8),
     SIPRES(8),SIENTH(8),SIMOLE(8),SICOMP(20,8)
COMMON/STMOUT/SONUM(8),SOFLAG(8),SOTYPE(8),SOVPFR(8),SOTEMP(8),
     SOPRES(8),SOENTH(8),SOMOLE(8),SOCOMP(20,8),SOKV(20,2)
```

Table A-11 Variable Definitions

EQPAR(I,J), I= 1, 25 - Equipment parameter vector for unit number J

NEMAX - maximum number of equipment pieces being used

MAXEQP - maximum length of EQPAR vector

NIN - number of input streams to the unit

NOUT - number of output streams from the unit

NOCOMP - number of components being used

NE - number of the unit currently being calculated

NEN - dummy equipment number

DERROR - maximum error for recycle convergence

LOOP - current number of loops attempted

LOOPS - maximum number of attempts allowed

KTRACE - printing option discussed in recycle information section

inlet and outlet properties (inlet,outlet):

SINUM,SONUM - system stream number

SIFLAG,SOFLAG - unit flag

SIVPFR,SOVPFR - vapor fraction

SITEMP,SOTEMP - temperature, R

SIPRES,SOPRES - stream pressure, psia

SIENTH,SOENTH - stream enthalpy, BTU/hr

SIMOLE,SOMOLE - total number of moles in the stream, lb moles/hr

SICOMP,SOCOMP(I,J) - moles of component I in stream number J

SOKV(I,J) - k-values of component I in stream number J

Table A-12 Thermodynamic Properties

BUBTP(ARG,ANS,LIST) - bubble point temperature

ENTH(ARG,ANS,0) - enthalpy at a given T & P

GETS(ARG,ANS,0) - entropy

KVAL(ARG,ANS,LIST) - equilibrium vaporization K-values

KVALUE(ARG,ANS,LIST) - equilibrium vaporization K-values for a liquid and a vapor stream

TSUBH(ARG,ANS,0) - temperature at given enthalpy and pressure

ZDENS(ARG,ANS,0) - vapor or liquid compressibility factor

simply include CALL ENTH(-1, ANS, 0). ANS will now be the enthalpy.

To use the flash routines, reset the EQPAR vector for that unit and call ADBF. The EQPAR vector should be exactly like EQPJ described in Section A-2 for that model with J being the unit number.

Once the model is written, it is added to CHESS through the DEC before it is compiled and sent to the CYBER. The typical student account on the DEC does not have enough memory to add a subroutine. This is because the editor needs to store a second copy of the program while editing.

The file OV5K contains all of the models, and towards the center, the lines

```
      OVERLAY(CHESS,5,17)
      PROGRAM FIVE17
C     SUBROUTINE AD12
      RETURN
      END
```

can be found. The model, AD12, is added between the lines 'C SUBROUTINE AD12' and 'RETURN'. There are similar sections for AD11 through ADD1.

CHESS should now be sent to the CYBER as discussed in Section A-5. Unfortunately, the overlay structure of CHESS does not allow the addition of subroutines once the program is compiled.

A-7 Example

This example demonstrates how the dealkylation of toluene was simulated using CHESS. The equipment flowsheet is shown in Figure 3.1. The CHESS flowsheet is Figure A-1. Table A-13 is the input data and Figure A-6 is the output data.

A-8 Common Errors

If the program did not work, check the dayfile for an error message. Some of the more common ones are discussed below.

'FILE NOT CATALOGUED,SN=SYSTEM' 'PF ABORT' means that the program could not be found. To solve this error, check the spelling of CHESS and ID.

FTN-FATAL ERROR 65 occurs when some part of the input data was left out. This includes the 7/8/9 card which separates the job information from the data.

FTN-FATAL ERROR 78 can result from several possible sources. These include giving an integer value for a real variable, giving a real value for in integer, omitting some data in the process configuration and component data section, and not inputting a comma at the end of a namelist entry.

FTN-FATAL ERROR 66 usually occurs when one of the strings $PMLIST, $EQLIST, $SEXLST, $SINLST, $KELIST, or $END was not started in column 2, when a namelist input was

started in column 1 so that the first letter of the variable name was ignored, or when more values for a vector were given than the vector was dimensioned for.

If the program is having trouble converging, try giving an initial estimate for the recycle stream, and make sure the duties are given for all heat exchangers, pumps, and compressors. The duty given in the input is the maximum that the unit can do. Therefore, if this is zero, or too small to achieve the specification indicated, the unit will not work properly.

The error 'DEW POINT CANNOT BE DETERMINED' is written by the thermodynamic routines when the dew point is below 200 R. This is not fatal unless the stream was supposed to be set to it's dew point. This error usually cannot be avoided when using hydrogen, so do not use a total condenser, or a flash to a dew point when using hydrogen.

Consult Appendix B in the FTN4 User's Manual for other errors found in the dayfile.

Table A-13 Input Data

```
CLEAN
 TOLUENE DEALKYLATION
 1MIXRM201   1 16 -2   0   0   0   0
 2PUMPP202   2 -3   0   0   0   0   0
 3PUMPP206  17 -4   0   0   0   0   0
 4MIXRM203   3  4 -5   0   0   0   0
 5HXERE204   5  8 -6 -9   0   0   0
 6FHTRF205   6 -7   0   0   0   0   0
 7REACR301   7 -8   0   0   0   0   0
 8ADBFD401   9-10-11   0   0   0   0
 9HXERE402  11-12   0   0   0   0   0
10DISCC403  12-13-14   0   0   0   0
11DVDRS501  14-15-16   0   0   0   0
12HXERE601  13-18   0   0   0   0   0
 0    0    0   0  0   0   0   0   0   0
 5
  1  41  40  46   2
 0
 0
 $EQLIST
 ENAME=1,4,48*0,
 EQP2=2.,0.,1000.,500.,-1.,20*0.,
 EQP3=3.,3.,1000.,500.,0.,20*0.,
 EQP5=5.,70.,0.,3.,1.,1.,3.,20.,5.,5.,15.,
 EQP6=6.,1000.,5.,1710.,19*0.,
 EQP7=7.,0.75,3.,-1.,-1.,0.,1.,17*0.,
 EQP8=8.,5.,600.,30.,21*0.,
 EQP9=9.,15.,0.,1.,1.,1.,5.,670.,5.,16*0.,
 EQP10=10.,2.,3.,0.99,2.,0.008,0.,0.,3.,16*0.,
 EQP11=11.,0.1,0.9,22*0.,
 EQP12=12.,90.,0.,1.,1.,1.,6.,530.,5.,16*0.,
 $END
 $SEXLST
 SNAME=2,3,4,5,6,7,9,10,11,12,13,14,15,18,86*0,
 SEX1=1.,0.,100.,0.,100.,0.,0.,16*0.,
 SEX8=8.,0.,481.,300.,25.,75.,6.,75.,15*0.,
 SEX16=16.,0.,25.6,1.,22.5,1.,0.1,1.,15*0.,
 SEX17=17.,0.,306.,300.,0.,0.,6.,0.,15*0.,
 $END
 $SINLST
 SNAME=2,3,4,5,6,7,9,11,12,13,14,89*0,
 SIN1=1.,1.,0.,0.,530.,14.696,4*0.,
 SIN8=8.,0.,0.,1.,1400.,490.,4*0.,
 SIN10=10.,2.,8*0.,
 SIN15=15.,2.,8*0.,
 SIN16=16.,0.,0.,0.,690.,25.,4*0.,
 SIN17=17.,1.,0.,1.,530.,100.,4*0.,
 SIN18=18.,2.,8*0.,
```

Table A-13 Continued

```
$END
$KELIST
LOOPS=20,
DERROR=0.01,
KE2=1,2,4,5,6,7,5,8,9,10,11,39*0,
$END
```

Figure A-6 Output Results

```
CLEAN
TOLUENE DEALKYLATION

NOW BEGIN TO READ PMLIST

 PMLIST READING COMPLETE

NOW BEGIN TO READ EQLIST

 EQLIST READING COMPLETE

NOW BEGIN TO READ SEXLST

 SEXLST READING COMPLETE

NOW BEGIN TO READ SINLST

 SINLST READING COMPLETE

NOW BEGIN TO READ KELIST

 KELIST READING COMPLETE
```

C H E S S

UNIVERSITY OF HOUSTON

CHEMICAL ENGINEERING SIMULATION SYSTEM

VERSION 3, JAN. 1978

C H E S S

TOLUENE DEALKYLATION

PROCESS VECTORS

..... EQUIPMENT ... ... NUMBER	SUBROUTINE	NAME	STREAM NUMBERS			
1	MIXR	M201	1	16	-2	0
2	PUMP	P202	2	-3	0	0
3	PUMP	P206	17	-4	0	0
4	MIXR	M203	3	4	-5	0
5	HXER	E204	5	8	-6	-9
6	FHTR	F205	6	-7	0	0
7	REAC	R301	7	-8	0	0
8	ADBF	D401	9	-10	-11	0
9	HXER	E402	11	-12	0	0
10	DISC	C403	12	-13	-14	0
11	DVDR	S501	14	-15	-16	0
12	HXER	E601	13	-18	0	0

TOLUENE DEALKYLATION

STREAM CONNECTIONS

STREAM	EQUIPMENT FROM	EQUIPMENT TO
1	0	1
2	1	2
3	2	4
4	3	4
5	4	5
6	5	6
7	6	7
8	7	5
9	5	8
10	8	0
11	8	9
12	9	10
13	10	12
14	10	11
15	11	0
16	11	1
17	0	3
18	12	0

TOLUENE DEALKYLATION

OTHER SYSTEM VARIABLES

NUMBER OF COMPONENTS	5
COMPONENT NUMBERS USED	1, 41, 40, 46, 2,
RECYCLE EQUIPMENT LIST (KE2)	1, 2, 4, 5, 6, 7, 5, 8, 9, 10, 11,
TOLERANCE, DERROR	.0100
MAX. LOOPS IN RECYCLE CALC.	20

"""INPUT DATA"""

TOLUENE DEALKYLATION

STREAM NUMBER	1	2	3	4
EQUIP. CONXION	FR 0 TO 1	FR 1 TO 2	FR 2 TO 4	FR 3 TO 4
VAPOR FRACTION	0.0000	0.0000	0.0000	0.0000
TEMPERATURE R	530.0000	0.0000	0.0000	0.0000
PRESSURE, PSIA	14.6960	0.0000	0.0000	0.0000
ENTHALPY,M BTU	-372.8023	0.0000	0.0000	0.0000
	COMPOSITION, LB-MOLES/HOUR			
HYDROGEN	0.0000	0.0000	0.0000	0.0000
TOLUENE	100.0000	0.0000	0.0000	0.0000
BENZENE	0.0000	0.0000	0.0000	0.0000
NITROGEN	0.0000	0.0000	0.0000	0.0000
METHANE	0.0000	0.0000	0.0000	0.0000
TOTAL	100.0000	0.0000	0.0000	0.0000

STREAM NUMBER	5	6	7	8
EQUIP. CONXION	FR 4 TO 5	FR 5 TO 6	FR 6 TO 7	FR 7 TO 5
VAPOR FRACTION	0.0000	0.0000	0.0000	1.0000
TEMPERATURE R	0.0000	0.0000	0.0000	1400.0000
PRESSURE, PSIA	0.0000	0.0000	0.0000	490.0000
ENTHALPY,M BTU	0.0000	0.0000	0.0000	10129.5902
	COMPOSITION, LB-MOLES/HOUR			
HYDROGEN	0.0000	0.0000	0.0000	300.0000
TOLUENE	0.0000	0.0000	0.0000	25.0000
BENZENE	0.0000	0.0000	0.0000	75.0000
NITROGEN	0.0000	0.0000	0.0000	6.0000
METHANE	0.0000	0.0000	0.0000	75.0000
TOTAL	0.0000	0.0000	0.0000	481.0000

STREAM NUMBER	9	10	11	12
EQUIP. CONXION	FR 5 TO 8	FR 8 TO 0	FR 8 TO 9	FR 9 TO 10
VAPOR FRACTION	0.0000	0.0000	0.0000	0.0000
TEMPERATURE R	0.0000	0.0000	0.0000	0.0000
PRESSURE, PSIA	0.0000	0.0000	0.0000	0.0000
ENTHALPY,M BTU	0.0000	0.0000	0.0000	0.0000
	COMPOSITION, LB-MOLES/HOUR			
HYDROGEN	0.0000	0.0000	0.0000	0.0000
TOLUENE	0.0000	0.0000	0.0000	0.0000
BENZENE	0.0000	0.0000	0.0000	0.0000
NITROGEN	0.0000	0.0000	0.0000	0.0000
METHANE	0.0000	0.0000	0.0000	0.0000
TOTAL	0.0000	0.0000	0.0000	0.0000

STREAM NUMBER	13	14	15	16
EQUIP. CONXION	FR 10 TO 12	FR 10 TO 11	FR 11 TO 0	FR 11 TO 1
VAPOR FRACTION	0.0000	0.0000	0.0000	.1937
TEMPERATURE R	0.0000	0.0000	0.0000	690.0000
PRESSURE, PSIA	0.0000	0.0000	0.0000	25.0000
ENTHALPY,M BTU	0.0000	0.0000	0.0000	138.6490
	COMPOSITION, LB-MOLES/HOUR			
HYDROGEN	0.0000	0.0000	0.0000	1.0000
TOLUENE	0.0000	0.0000	0.0000	22.5000
BENZENE	0.0000	0.0000	0.0000	1.0000
NITROGEN	0.0000	0.0000	0.0000	.1000
METHANE	0.0000	0.0000	0.0000	1.0000
TOTAL	0.0000	0.0000	0.0000	25.6000

STREAM NUMBER	17	18
EQUIP. CONXION	FR 0 TO 3	FR 12 TO 0
VAPOR FRACTION	1.0000	0.0000
TEMPERATURE R	530.0000	0.0000
PRESSURE, PSIA	100.0000	0.0000
ENTHALPY,M BTU	2531.5834	0.0000
	COMPOSITION, LB-MOLES/HOUR	
HYDROGEN	300.0000	0.0000
TOLUENE	0.0000	0.0000
BENZENE	0.0000	0.0000
NITROGEN	6.0000	0.0000
METHANE	0.0000	0.0000
TOTAL	306.0000	0.0000

INPUT DATA

TOLUENE DEALKYLATION

* EQUIPMENT SUMMARY - EQUIPMENT LIST *

EQ. #	EXT. NAME	SUB. NAME
1	M201	MIXR
2	P202	PUMP
3	P206	PUMP
4	M203	MIXR
5	E204	HXER
6	F205	FHTR
7	R301	REAC
8	D401	ADBF
9	E402	HXER
10	C403	DISC
11	S501	DVDR
12	E601	HXER

TOLUENE DEALKYLATION

* EQUIPMENT SUMMARY - INDIVIDUAL DETAILS *

***DIVIDERS ***

EQUIPMENT NO.		11
EXTERNAL NAME		S501
FRXN. #	1	.1000
	2	.9000
	3	0.0000
	4	0.0000
	5	0.0000
	6	0.0000

.....

***MIXERS-W/FLASH ***

EQUIPMENT NO.	1	4
EXTERNAL NAME	M201	M203

.....

***GENERAL FLASH UNIT3 ***

EQUIPMENT NO.	8
EXTERNAL NAME	D401
MODE	5.0000
PARAM # 3	500.0000
PARAM # 4	30.0000
HEAT DUTY,MBTU	0.0000
K-VALUE # 1	0.0000
K-VALUE # 2	0.0000
K-VALUE # 3	0.0000
K-VALUE # 4	0.0000
K-VALUE # 5	0.0000

.....

***REACTORS ***

EQUIPMENT NO.	7
EXTERNAL NAME	R301
KEY COMP CONV	.7500
KEY COMP #	2.0000
STOICH. FAC. 1	-1.0000

2	-1.0000
3	1.0000
4	0.0000
5	1.0000

.....

EXCHANGER/CONDENSERS

EQUIPMENT NO.	5	9	12
EXTERNAL NAME	E204	E402	E601
U	70.0000	15.0000	90.0000
AREA	0.0000	0.0000	0.0000
# SHELLS	3.0000	1.0000	1.0000
SHELL PASSES	1.0000	1.0000	1.0000
TUBE PASSES	1.0000	1.0000	1.0000
MODE	3.0000	5.0000	6.0000
DT OR T-OUT	20.0000	670.0000	530.0000
DELTA P-STM 1	5.0000	5.0000	5.0000
DELTA P-STM 2	5.0000	0.0000	0.0000
Q-STREAM 1 (M BTU/HR)	0.0000	0.0000	0.0000
WATER USAGE (GAL/HR)	0.0000	0.0000	0.0000

.....

***PUMPS/COMPRESSORS ***

EQUIPMENT NO.	2	3
EXTERNAL NAME	P202	P206
COMP. STAGES	0.0000	3.0000
WORK CAPACITY (M BTU/HR)	1000.0000	1000.0000
OUTLET PRES. (PSIA)	500.0000	500.0000
POWER TYPE : (+)-STEAM (0)-ELEC. (-)-FUEL GAS	-1.0000	0.0000
H-OUTLET STEAM (BTU/LB)	0.0000	0.0000
FUEL USAGE (MSCF/HR)	0.0000	0.0000
WATER USAGE (GAL/HR)	0.0000	0.0000
STEAM USAGE (M LBS/HR)	0.0000	0.0000
KW USAGE	0.0000	0.0000

.....

***FIRED HEATERS ***

EQUIPMENT NO.	6
EXTERNAL NAME	F205
HEAT DUTY (M BTU/HR)	1000.0000
DELTA PRES. (PSIA)	5.0000
TEMP. OUT (DEG R.)	1400.0000
Q ABSORBED (M BTU/HR)	0.0000
FUEL USAGE (MSCF/HR)	0.0000

.....

SHORT-CUT DISTILL'NS

EQUIPMENT NO.	10
EXTERNAL NAME	C403
OPTION	2.0000
LIGHT KEY(LK)	3.0000
FRACTION LK	.9900
HEAVY KEY(HK)	2.0000
FRACTION HK	.0080
CONDENSER	0.0000
REFLUX RATIO(R)	0.0000
R/RM	3.0000
STAGE NUMBER(S)	0.0000
S-MIN	0.0000
S-FEED	0.0000
QC (M BTU/HR)	0.0000
QR (M BTU/HR)	0.0000

.....

```
*****BEGIN TRIAL AND ERROR RECYCLE CALCULATIONS WITH
EQUIPMENT LIST..
   1, 2, 4, 5, 6, 7, 5, 8, 9, 10, 11,

 ... BEGIN LOOP 1 ...

***** AFLASH DID NOT CONVERGE, UNIT NO.- 1

***    DEW POINT TEMP. CANNOT BE DETERMINED, UNIT NO.  5

***    DEW POINT TEMP. CANNOT BE DETERMINED, UNIT NO.  6

***    DEW POINT TEMP. CANNOT BE DETERMINED, UNIT NO.  5

***TEMPERATURE BOUNDED AT 200R IN TSUBH, DEWTP OR BUBTP

***BUBBLE POINT TEMP. CANNOT BE DETERMINED, UNIT NO. 10

 ... BEGIN LOOP 2 ...

***    DEW POINT TEMP. CANNOT BE DETERMINED, UNIT NO.  5

***    DEW POINT TEMP. CANNOT BE DETERMINED, UNIT NO.  6

***    DEW POINT TEMP. CANNOT BE DETERMINED, UNIT NO.  5

***TEMPERATURE BOUNDED AT 200R IN TSUBH, DEWTP OR BUBTP

***BUBBLE POINT TEMP. CANNOT BE DETERMINED, UNIT NO. 10

 ... BEGIN LOOP 3 ...

***    DEW POINT TEMP. CANNOT BE DETERMINED, UNIT NO.  5

***    DEW POINT TEMP. CANNOT BE DETERMINED, UNIT NO.  5

***TEMPERATURE BOUNDED AT 200R IN TSUBH, DEWTP OR BUBTP

***BUBBLE POINT TEMP. CANNOT BE DETERMINED, UNIT NO. 10

 ... BEGIN LOOP 4 ...

***    DEW POINT TEMP. CANNOT BE DETERMINED, UNIT NO.  5

***    DEW POINT TEMP. CANNOT BE DETERMINED, UNIT NO.  5

***TEMPERATURE BOUNDED AT 200R IN TSUBH, DEWTP OR BUBTP

***BUBBLE POINT TEMP. CANNOT BE DETERMINED, UNIT NO. 10

***** END OF RECYCLE CALCULATIONS
```

""""FINAL RESULTS""""

TOLUENE DEALKYLATION

STREAM NUMBER	1	2	3	4
EQUIP. CONXION	FR 0 TO 1	FR 1 TO 2	FR 2 TO 4	FR 3 TO 4
VAPOR FRACTION	0.0000	0.0000	0.0000	1.0000
TEMPERATURE R	530.0000	575.2618	575.2618	530.0000
PRESSURE, PSIA	14.6960	14.6960	500.0000	500.0000
ENTHALPY,M BTU	-372.8023	-225.5742	-211.1267	2535.1221
	COMPOSITION, LB-MOLES/HOUR			
HYDROGEN	0.0000	.0000	.0000	300.0000
TOLUENE	100.0000	127.7358	127.7358	0.0000
BENZENE	0.0000	.7975	.7975	0.0000
NITROGEN	0.0000	.0000	.0000	6.0000
METHANE	0.0000	.0000	.0000	0.0000
TOTAL	100.0000	128.5333	128.5333	306.0000

STREAM NUMBER	5	6	7	8
EQUIP. CONXION	FR 4 TO 5	FR 5 TO 6	FR 6 TO 7	FR 7 TO 5
VAPOR FRACTION	.6999	1.0000	1.0000	1.0000
TEMPERATURE R	560.4684	1327.7457	1400.0000	1400.0000
PRESSURE, PSIA	500.0000	495.0000	490.0000	490.0000
ENTHALPY,M BTU	2323.9954	10024.5904	10692.0888	10342.8266
	COMPOSITION, LB-MOLES/HOUR			
HYDROGEN	300.0000	300.0000	0.0000	300.0000
TOLUENE	127.7358	127.7358	127.7358	31.9340
BENZENE	.7975	.7975	.7975	96.5994
NITROGEN	6.0000	6.0000	6.0000	6.0000
METHANE	.0000	.0000	.0000	95.8019
TOTAL	434.5333	434.5333	434.5333	434.5333

STREAM NUMBER	9	10	11	12
EQUIP. CONXION	FR 5 TO 8	FR 8 TO 0	FR 8 TO 9	FR 9 TO 10
VAPOR FRACTION	.7006	1.0000	0.0000	.0156
TEMPERATURE R	580.4684	500.0000	500.0000	670.0000
PRESSURE, PSIA	485.0000	30.0000	30.0000	25.0000
ENTHALPY,M BTU	2642.2317	2647.2523	-502.5885	294.5466
	COMPOSITION, LB-MOLES/HOUR			
HYDROGEN	204.1981	204.1283	.0698	.0698
TOLUENE	31.9340	.8163	31.1177	31.1177
BENZENE	96.5994	7.8257	88.7736	88.7736
NITROGEN	6.0000	5.9872	.0128	.0128
METHANE	95.8019	95.5023	.2995	.2995
TOTAL	434.5333	314.2598	120.2735	120.2735

STREAM NUMBER	13	14	15	16
EQUIP. CONXION	FR 10 TO 12	FR 10 TO 11	FR 11 TO 0	FR 11 TO 1
VAPOR FRACTION	1.0000	0.0000	0.0000	0.0000
TEMPERATURE R	671.0463	729.9663	729.9663	729.9663
PRESSURE, PSIA	25.0000	25.0000	25.0000	25.0000
ENTHALPY,M BTU	1263.5987	163.8585	16.3858	147.4726
	COMPOSITION, LB-MOLES/HOUR			
HYDROGEN	.0698	.0000	.0000	0.0000
TOLUENE	.2489	30.8687	3.0869	27.7819
BENZENE	87.8859	.8877	.0888	.7990
NITROGEN	.0128	.0000	.0000	.0000
METHANE	.2995	.0000	.0000	.0000
TOTAL	88.5170	31.7565	3.1756	28.5808

STREAM NUMBER	17	18
EQUIP. CONXION	FR 0 TO 3	FR 12 TO 0
VAPOR FRACTION	1.0000	0.0018
TEMPERATURE R	530.0000	530.0000
PRESSURE, PSIA	100.0000	20.0000
ENTHALPY,M BTU	2531.5834	-251.4824
	COMPOSITION, LB-MOLES/HOUR	
HYDROGEN	300.0000	.0698
TOLUENE	0.0000	.2489
BENZENE	0.0000	87.8859
NITROGEN	6.0000	.0128
METHANE	0.0000	.2995
TOTAL	306.0000	88.5170

FINAL RESULTS

TOLUENE DEALKYLATION

* EQUIPMENT SUMMARY - EQUIPMENT LIST *

EQ. #	EXT. NAME	SUB. NAME
1	M201	MIXR
2	P202	PUMP
3	P206	PUMP
4	M203	MIXR
5	E204	HXER
6	F205	FHTR
7	R301	REAC
8	D401	ADBF
9	E402	HXER
10	C403	DISC
11	S501	DVDR
12	E601	HXER

TOLUENE DEALKYLATION

* EQUIPMENT SUMMARY - INDIVIDUAL DETAILS *

***DIVIDERS ***

EQUIPMENT NO.		11
EXTERNAL NAME		S501
FRXN. #	1	.1000
	2	.9000
	3	0.0000
	4	0.0000
	5	0.0000
	6	0.0000

.....

***MIXERS-W/FLASH ***

EQUIPMENT NO.	1	4
EXTERNAL NAME	M201	M203

.....

***GENERAL FLASH UNIT3 ***

EQUIPMENT NO.		8
EXTERNAL NAME		D401
MODE		5.0000
PARAM #	3	500.0000
PARAM #	4	30.0000
HEAT DUTY,MBTU		-497.5679
K-VALUE #	1	1118.7098
K-VALUE #	2	.0100
K-VALUE #	3	.0337
K-VALUE #	4	178.8530
K-VALUE #	5	122.0209

.....

***REACTORS ***

EQUIPMENT NO.		7
EXTERNAL NAME		R301
KEY COMP CONV		.7500
KEY COMP #		2.0000
STOICH. FAC.	1	-1.0000

2	-1.0000
3	1.0000
4	0.0000
5	1.0000

.....

EXCHANGER/CONDENSERS

EQUIPMENT NO.	5	9	12
EXTERNAL NAME	E204	E402	E601
U	70.0000	15.0000	90.0000
AREA	901.3704	0.0000	0.0000
# SHELLS	3.0000	1.0000	1.0000
SHELL PASSES	1.0000	1.0000	1.0000
TUBE PASSES	1.0000	1.0000	1.0000
MODE	3.0000	5.0000	6.0000
DT OR T-OUT	20.0000	670.0000	530.0000
DELTA P-STM 1	5.0000	5.0000	5.0000
DELTA P-STM 2	5.0000	0.0000	0.0000
Q-STREAM 1 (M BTU/HR)	-7700.5949	-797.1352	-1315.4635
WATER USAGE (GAL/HR)	0.0000	-6379.6331	-10527.9193

.....

***PUMPS/COMPRESSORS ***

EQUIPMENT NO.	2	3
EXTERNAL NAME	P202	P206
COMP. STAGES	0.0000	3.0000
WORK CAPACITY (M BTU/HR)	1000.0000	1000.0000
OUTLET PRES. (PSIA)	500.0000	500.0000
POWER TYPE : (+)-STEAM (0)-ELEC. (-)-FUEL GAS	-1.0000	0.0000
H-OUTLET STEAM (BTU/LB)	0.0000	0.0000
FUEL USAGE (MSCF/HR)	70.1697	0.0000
WATER USAGE (GAL/HR)	44.6592	4424.5989
STEAM USAGE (M LBS/HR)	0.0000	0.0000
KW USAGE	0.0000	191.6605

.....

***FIRED HEATERS ***

EQUIPMENT NO.	6
EXTERNAL NAME	F205
HEAT DUTY (M BTU/HR)	1000.0000
DELTA PRES. (PSIA)	5.0000
TEMP. OUT (DEG R.)	1400.0000
Q ABSORBED (M BTU/HR)	677.2783
FUEL USAGE (MSCF/HR)	1.0034

.....

SHORT-CUT DISTILL'NS

EQUIPMENT NO.	10
EXTERNAL NAME	C403
OPTION	2.0000
LIGHT KEY(LK)	3.0000
FRACTION LK	.9900
HEAVY KEY(HK)	2.0000
FRACTION HK	.0080
CONDENSER	0.0000
REFLUX RATIO(R)	3.6284
R/RM	3.0000
STAGE NUMBER(S)	14.2513
S-MIN	11.5109
S-FEED	7.2930
QC (M BTU/HR)	13100.9993
QR (M BTU/HR)	11403.3652

.....

APPENDIX B

CHESS EXAMPLES

These are the examples mentioned in Chapter 4.

1. Cyclopentadiene Recovery Unit

A. Input Data

```
CLEAN
 CHESS EX 2 - CYCLOPENTADIENE RECOVERY UNIT
 1PUMPPUMP   1 -2   0   0   0   0   0
 2FHTRHX1    2 -3   0   0   0   0   0
 3REACRX1    3 -4   0   0   0   0   0
 4REACRX2    4 -5   0   0   0   0   0
 5HXERHX2    5 -6   0   0   0   0   0
 6MSEQDC1   13 18   9 16 -7-11   0
 7OVHDCOND   7-10 -9   0   0   0   0
10BOTTREB   11-13-14   0   0   0   0
11REACRX3   14-19   0   0   0   0   0
12HXERHX5   15-17   0   0   0   0   0
13VALVV1     6-18   0   0   0   0   0
14BOTTHX6   19-16-15   0   0   0   0
 0    0    0   0   0   0   0   0   0   0
 5
 34 34   7 14 14
 1
 CPD
 5
 TP
 9
 IP
13
 DC4
17
 COD
 0
 0
 $EQLIST
 EQP1=1.,1.,12.,125.,21*0.,
 EQP2=2.,5000.,0.,654.,21*0.,
 EQP3=3.,.925,1.,-1.,0.,0.,.5,18*0.,
 EQP4=4.,.075,2.,2-1.,0.,0.,1.,17*0.,
 EQP5=5.,25.,100.,3*1.,6.,700.,17*0.,
 EQP6=6.,4.,4.,2.,1.,4.,19*0.,
```

```
EQP7=7.,1.,.4,22*0.,
EQP10=10.,3.,4.,22*0.,
EQP11=11.,.03,4.,0.5,0.,0.,-0.5,18*0.,
EQP12=12.,50.,40.,1.,1.,1.,4.,18*0.,
EQP13=13.,25.,23*0.,
EQP14=14.,3.,1.,22*0.,
$END
$SEXLST
SNAME=2,3,4,5,6,7,9,10,11,13,14,15,16,17,18,19,84*0,
SEX1=1.,2360.,144.4,30.6,15.,1.,97.8,16*0.,
$END
$SINLST
SNAME=2,3,4,5,6,7,9,10,11,13,14,15,16,17,18,19,84*0,
SIN1=1.,1.,0.,0.,560.,115.,4*0.,
SIN10=10.,2.,8*0.,
SIN17=17.,2.,8*0.,
$END
```

B. Partial Output Results

CHESS EX 2 - CYCLOPENTADIENE RECOVERY UNIT

PROCESS VECTORS

..... EQUIPMENT			STREAM NUMBERS					
NUMBER	SUBROUTINE	NAME						
1	PUMP	PUMP	1	-2	0	0	0	0
2	FHTR	HX1	2	-3	0	0	0	0
3	REAC	RX1	3	-4	0	0	0	0
4	REAC	RX2	4	-5	0	0	0	0
5	HXER	HX2	5	-6	0	0	0	0
6	MSEQ	DC1	13	18	9	16	-7	-11
7	OVHD	COND	7	-10	-9	0	0	0
10			11	-13	-14	0	0	0
11			14	-19	0	0	0	0
12	BOTT	REB	15	-17	0	0	0	0
13	REAC	RX3	6	-18	0	0	0	0
14	HXER	HX5	19	-16	-15	0	0	0

(Note: The names have been shifted and no longer correspond to the correct stream information)

CHESS EX 2 - CYCLOPENTADIENE RECOVERY UNIT

STREAM CONNECTIONS

STREAM	EQUIPMENT FROM	EQUIPMENT TO
1	0	1
2	1	2
3	2	3
4	3	4
5	4	5
6	5	13
7	6	7
9	7	6
10	7	0
11	6	10
13	10	6
14	10	11
15	14	12
16	14	6
17	12	0
18	13	6
19	11	14

(The stream information is correct)

(Skipping to the Equipment Summaries for the input data)

INPUT DATA

CHESS EX 2 - CYCLOPENTADIENE RECOVERY UNIT

* EQUIPMENT SUMMARY - EQUIPMENT LIST *

EQ. #	EXT. NAME	SUB. NAME
1	PUMP	PUMP
2	HX1	FHTR
3	RX1	REAC
4	RX2	REAC
5	HX2	HXER
6	DC1	MSEQ
7	COND	OVHD
10	REB	BOTT
11	RX3	REAC
12	HX5	HXER

(Final two units have been dropped. The simulation continued as usual from here without converging since some of the streams were never given or calculated)

C. Revised Input Data

```
CLEAN
 CHESS EX 2 - CYCLOPENTADIENE RECOVERY UNIT
 1PUMPPUMP   1 -2   0   0   0   0   0
 2FHTRHX1    2 -3   0   0   0   0   0
 3REACRX1    3 -4   0   0   0   0   0
 4REACRX2    4 -5   0   0   0   0   0
 5HXERHX2    5 -6   0   0   0   0   0
 6MSEQDC1   13 18   9 16 -7-11   0
 7OVHDCOND   7-10 -9   0   0   0   0
 8BOTTREB   11-13-14   0   0   0   0
 9REACRX3   14-19   0   0   0   0   0
10HXERHX5   15-17   0   0   0   0   0
11VALVV1     6-18   0   0   0   0   0
12BOTTHX6   19-16-15   0   0   0   0
 0    0    0   0   0   0   0   0   0   0
 5
 34 34   7 14 14
 1
 CPD
 5
 TP
 9
 IP
13
 DC4
17
 COD
 0
 0
 $EQLIST
 EQP1=1.,1.,12.,125.,21*0.,
 EQP2=2.,5000.,0.,654.,21*0.,
 EQP3=3.,.925,1.,-1.,0.,0.,.5,18*0.,
 EQP4=4.,.075,2.,2-1.,0.,0.,1.,17*0.,
 EQP5=5.,25.,100.,3*1.,6.,700.,17*0.,
 EQP6=6.,4.,4.,2.,1.,4.,19*0.,
 EQP7=7.,1.,.4,22*0.,
 EQP8=8.,3.,4.,22*0.,
 EQP9=9.,.03,4.,0.5,0.,0.,-0.5,18*0.,
 EQP10=10.,50.,40.,1.,1.,1.,4.,18*0.,
 EQP11=11.,25.,23*0.,
 EQP12=12.,3.,1.,22*0.,
 $END
 $SEXLST
 SNAME=2,3,4,5,6,7,9,10,11,13,14,15,16,17,18,19,84*0,
 SEX1=1.,2360.,144.4,30.6,15.,1.,97.8,16*0.,
 $END
 $SINLST
 SNAME=2,3,4,5,6,7,9,10,11,13,14,15,16,17,18,19,84*0,
```

```
SIN1=1.,1.,0.,0.,560.,115.,4*0.,
SIN10=10.,2.,8*0.,
SIN17=17.,2.,8*0.,
$END
$KELIST
KE4=11,9*0,LOOPS=40,DERROR=0.01,KE2=6,7,8,9,12,45*0,
$END
```

D. New Partial Output Results

CHESS EX 2 - CYCLOPENTADIENE RECOVERY UNIT

PROCESS VECTORS

..... EQUIPMENT			STREAM NUMBERS					
NUMBER	SUBROUTINE	NAME						
1	PUMP	PUMP	1	-2	0	0	0	0
2	FHTR	HX1	2	-3	0	0	0	0
3	REAC	RX1	3	-4	0	0	0	0
4	REAC	RX2	4	-5	0	0	0	0
5	HXER	HX2	5	-6	0	0	0	0
6	MSEQ	DC1	13	18	9	16	-7	-11
7	OVHD	COND	7	-10	-9	0	0	0
8	BOTT	REB	11	-13	-14	0	0	0
9	REAC	RX3	14	-19	0	0	0	0
10	HXER	HX5	15	-17	0	0	0	0
11	VALV	V1	6	-18	0	0	0	0
12	BOTT	HX6	19	-16	-15	0	0	0

CHESS EX 2 - CYCLOPENTADIENE RECOVERY UNIT

STREAM CONNECTIONS

STREAM	EQUIPMENT FROM	EQUIPMENT TO
1	0	1
2	1	2
3	2	3
4	3	4
5	4	5
6	5	11
7	6	7
9	7	6
10	7	0
11	6	8
13	8	6
14	8	9
15	12	10
16	12	6
17	10	0
18	11	6
19	9	12

(Skipping to the Equipment Summaries for the input data)

INPUT DATA

CHESS EX 2 - CYCLOPENTADIENE RECOVERY UNIT

* EQUIPMENT SUMMARY - EQUIPMENT LIST *

EQ. #	EXT. NAME	SUB. NAME
1	PUMP	PUMP
2	HX1	FHTR
3	RX1	REAC
4	RX2	REAC
5	HX2	HXER
6	DC1	MSEQ
7	COND	OVHD
8	REB	BOTT
9	RX3	REAC
10	HX5	HXER
11	V1	VALV
12	HX6	BOTT

2. Controlled Reboiler

A. Input Data

```
CLEAN
 CHESS EX5 - STRIPPER - REBOILER
 1ABSRSTPR  4   1 -2 -3   0   0   0
 2CTRLREBL  3 -4 -5  0   0   0   0
 0    0    0   0   0   0   0   0   0   0
10
 49 46   2   3   4   5   6   7   8 11
  0
  0
 $EQLIST
 EQP1=1.,10.,80.,22*0.,
 EQP2=2.,1.,12*0.,1.,31.,.02,1.,2.,3.,4.,.001,630.,615.,0.,
 $END
 $SEXLST
 SNAME=1,2,3,4,5,95*0,
 SEX1=1.,0.,95.365,.088,.149,28.786,20.103,24.977,2.832,
     10.221,2.224,2.396,3.589,10*0.,
 $END
 $SINLST
 SNAME=1,2,3,4,5,95*0,
 SIN1=1.,1.,0.,0.,430.,415.,4*0.,
 SIN2=2.,2.,8*0.,
 SIN5=5.,2.,8*0.,
 $END
 $KELIST
 KE2=1,2,48*0,DERROR=0.01,LOOPS=25,KE4=3,4,8*0,
 $END
```

B. Partial Output Results

CHESS EX5 - STRIPPER - REBOILER

PROCESS VECTORS

..... EQUIPMENT			STREAM NUMBERS			
NUMBER	SUBROUTINE	NAME				
1	ABSR	STPR	4	1	-2	-3
2	CTRL	REBL	3	-4	-5	0

CHESS EX5 - STRIPPER - REBOILER

STREAM CONNECTIONS

STREAM	EQUIPMENT FROM	EQUIPMENT TO
1	0	1
2	1	0
3	1	2
4	2	1
5	2	0

CHESS EX5 - STRIPPER - REBOILER

OTHER SYSTEM VARIABLES

NUMBER OF COMPONENTS	10
COMPONENT NUMBERS USED	49, 46, 2, 3, 4, 5, 6, 7, 8, 11,
RECYCLE EQUIPMENT LIST (KE2)	1, 2,
STREAMS USED IN CONV. ROUTINE (KE4)	3, 4,
TOLERANCE, DERROR	0.0100
MAX. LOOPS IN RECYCLE CALC.	25

"""INPUT DATA"""

CHESS EX5 - STRIPPER - REBOILER

STREAM NUMBER	1	2	3	4
EQUIP. CONXION	FR 0 TO 1	FR 1 TO 0	FR 1 TO 2	FR 2 TO 1
VAPOR FRACTION	0.0000	0.0000	0.0000	0.0000
TEMPERATURE R	430.0000	0.0000	0.0000	0.0000
PRESSURE, PSIA	415.0000	0.0000	0.0000	0.0000
ENTHALPY,M BTU	295.4472	0.0000	0.0000	0.0000
	COMPOSITION, LB-MOLES/HOUR			
CO2	.0880	0.0000	0.0000	0.0000
NITROGEN	.1490	0.0000	0.0000	0.0000
METHANE	28.7860	0.0000	0.0000	0.0000
ETHANE	20.1030	0.0000	0.0000	0.0000
PROPANE	24.9770	0.0000	0.0000	0.0000
I-BUTANE	2.8320	0.0000	0.0000	0.0000
N-BUTANE	10.2210	0.0000	0.0000	0.0000
I-PENTANE	2.2240	0.0000	0.0000	0.0000
N-PENTANE	2.3960	0.0000	0.0000	0.0000
N-HEPTANE	3.5890	0.0000	0.0000	0.0000
TOTAL	95.3650	0.0000	0.0000	0.0000

STREAM NUMBER	5
EQUIP. CONXION	FR 2 TO 0
VAPOR FRACTION	0.0000
TEMPERATURE R	0.0000
PRESSURE, PSIA	0.0000
ENTHALPY,M BTU	0.0000
	COMPOSITION, LB-MOLES/HOUR
CO2	0.0000
NITROGEN	0.0000
METHANE	0.0000
ETHANE	0.0000
PROPANE	0.0000
I-BUTANE	0.0000
N-BUTANE	0.0000
I-PENTANE	0.0000
N-PENTANE	0.0000
N-HEPTANE	0.0000
TOTAL	0.0000

INPUT DATA

CHESS EX 5 - STRIPPER - REBOILER

* EQUIPMENT SUMMARY - EQUIPMENT LIST *

EQ. #	EXT. NAME	SUB. NAME
1	STPR	ABSR
2	REBL	CTRL

CHESS EX5 - STRIPPER - REBOILER

* EQUIPMENT SUMMARY - INDIVIDUAL DETAILS *

***ABSORBERS ***

EQUIPMENT NO. 1

EXTERNAL NAME	STPR
# OF STAGES	10.0000
DELTA T. (DEG R.)	80.0000
MODE	0.0000
Q ABSORBED (M BTU/HR)	0.0000

.....

***MISC. CONTROLLERS ***

EQUIPMENT NO. 2

EXTERNAL NAME	REBL
PARAMETERS :	1.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
CONTROL OBJ.	1.0000
MODE	31.0000
CONT. VALUE	.0200
STREAM IN	1.0000
STREAM OUT	2.0000
COMPONENT A	3.0000
COMPONENT B	4.0000
TOLERANCE	.0010
UPPER LIMIT	630.0000
LOWER LIMIT	615.0000

.....

```
*****BEGIN TRIAL AND ERROR RECYCLE CALCULATIONS WITH
EQUIPMENT LIST..  1, 2,

 ... BEGIN  LOOP   1 ...

*****    ABSR   BY-PASSED, ZERO INPUT(S) EXISTING, NE= 1
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   2 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   3 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   4 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   5 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   6 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   7 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   8 ...

***   DEW POINT TEMP. CANNOT BE DETERMINED, UNIT NO.  1
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   9 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   10 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   11 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   12 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   13 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   14 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   15 ...
***** CTRL     2 DID NOT CONVERGE.
```

```
 ... BEGIN  LOOP   16 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   17 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   18 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   19 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   20 ...
***** CTRL     2 DID NOT CONVERGE.

 ... BEGIN  LOOP   21 ...
***** CTRL     2 DID NOT CONVERGE.

***** END OF RECYCLE CALCULATIONS
```

"""FINAL RESULTS"""

CHESS EX5 - STRIPPER - REBOILER

STREAM NUMBER	1	2	3	4
EQUIP. CONXION	FR 0 TO 1	FR 1 TO 0	FR 1 TO 2	FR 2 TO 1
VAPOR FRACTION	0.0000	1.0000	0.0000	1.0000
TEMPERATURE R	430.0000	467.5174	525.6890	617.4895
PRESSURE, PSIA	415.0000	415.0000	415.0000	415.0000
ENTHALPY,M BTU	295.4472	318.9384	486.0590	508.6538
	COMPOSITION, LB-MOLES/HOUR			
CO2	.0880	.0340	.1544	.1002
NITROGEN	.1490	.1480	.0029	.0019
METHANE	28.7860	28.6128	.7979	.6321
ETHANE	20.1030	6.1054	34.4370	20.4058
PROPANE	24.9770	1.8988	39.5725	16.4531
I-BUTANE	2.8320	.0796	3.8548	1.0995
N-BUTANE	10.2210	.1971	13.2566	3.2237
I-PENTANE	2.2240	.0176	2.6015	.3940
N-PENTANE	2.3960	.0137	2.7406	.3573
N-HEPTANE	3.5890	.0022	3.7111	.1239
TOTAL	95.3650	37.1091	101.1293	42.7914

STREAM NUMBER	5
EQUIP. CONXION	FR 2 TO 0
VAPOR FRACTION	0.0000
TEMPERATURE R	617.4895
PRESSURE, PSIA	415.0000
ENTHALPY,M BTU	443.4270
	COMPOSITION, LB-MOLES/HOUR
CO2	.0541
NITROGEN	.0010
METHANE	.1658
ETHANE	14.0312
PROPANE	23.1194
I-BUTANE	2.7553
N-BUTANE	10.0329
I-PENTANE	2.2075
N-PENTANE	2.3833
N-HEPTANE	3.5872
TOTAL	58.3379

FINAL RESULTS

CHESS EX 5 - STRIPPER - REBOILER

* EQUIPMENT SUMMARY - EQUIPMENT LIST *

EQ. #	EXT. NAME	SUB. NAME
1	STPR	ABSR
2	REBL	CTRL

CHESS EX5 - STRIPPER - REBOILER

* EQUIPMENT SUMMARY - INDIVIDUAL DETAILS *

***ABSORBERS ***

EQUIPMENT NO. 1

EXTERNAL NAME	STPR
# OF STAGES	10.0000
DELTA T. (DEG R.)	80.0000
MODE	0.0000
Q ABSORBED (M BTU/HR)	-.1511

.....

***MISC. CONTROLLERS ***

EQUIPMENT NO. 2

EXTERNAL NAME	REBL
PARAMETERS :	1.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
	0.0000
CONTROL OBJ.	1.0000
MODE	31.0000
CONT. VALUE	.0200
STREAM IN	1.0000
STREAM OUT	2.0000
COMPONENT A	3.0000
COMPONENT B	4.0000
TOLERANCE	.0010
UPPER LIMIT	630.0000
LOWER LIMIT	615.0000

.....

APPENDIX C

PROCESS INSTALLATION

This Appendix contains the commands to install the PROCESS program on a CDC CYBER 175. The procedure is best done in steps.

Step 1. Unload the files from the tape.

```
REQUEST,TAPE,NT,PE,RO,VSN=7184B.
COPYBF,TAPE,UNLOAD.
RETURN,UNLOAD.
REQUEST,SEGPRO,PF.
COPYBF,TAPE,SEGPRO.
CATALOG,SEGPRO,SEGPRO,ID=SLD,RP=1.
RETURN,SEGPRO.
REQUEST,PREPPL,PF.
COPYBF,TAPE,PREPPL.
CATALOG,PREPPL,PREPPL,ID=SLD,RP=1.
RETURN,PREPPL.
REQUEST,PROCPL,PF.
COPYBF,TAPE,PROCPL.
CATALOG,PROCPL,PROCPL,ID=SLD,RP=1.
RETURN,PROCPL.
REQUEST,PROLIB,PF.
COPYBF,TAPE,PROLIB.
```

```
CATALOG,PROLIB,PROLIB,ID=SLD,RP=1.
```

Step 2. Send the file PROCPL through UPDATE to get a FORTRAN listing of the PROCESS program. The FORTRAN listing will appear in the file COMPILE.

```
ATTACH,PROCPL,ID=SLD.
UPDATE,A,P=PROCPL,O=0.
REWIND,NEWPL.
REQUEST,COMPILE,PF.
UPDATE,F,P=NEWPL,L=A1,I=0.
CATALOG,COMPILE,COMPILE,ID=SLD,RP=1.
PURGE,PROCPL,ID=SLD.
```

Step 3. Compile the PROCESS program.

```
ATTACH,COMPILE,ID=SLD.
REQUEST,PROBN,PF.
FTN,I=COMPILE,OPT=2,L=0.
REWIND,LGO.
COPYBF,LGO,PROBN.
CATALOG,PROBN,PROBN,ID=SLD,RP=1.
PURGE,COMPILE,ID=SLD.
```

Step 4. Load the PROCESS program. This step is separated from the compilation of the PROCESS program to conserve memory. This step does not work.

```
REQUEST,PROCES,PF.
ATTACH,PROBN,ID=SLD.
ATTACH,SEGPRO,ID=SLD.
SEGLOAD,B=PROCES,I=SEGPRO.
```

```
LOAD,PROBN.
NOGO,PROCES.
CATALOG,PROCES,PROCES,ID=SLD,RP=1.
```

Step 5. Send the file PREPPL through UPDATE to get a FORTRAN listing of the PREPRO program. The PREPRO program contains an error and some calls to non-existent internal functions, which must also be changed during the UPDATE program. The FORTRAN listing will appear in the file COMP.

```
ATTACH,PREPPL,ID=SLD.
REQUEST,NEWPL,PF.
UPDATE,F,P=PREPPL,O=0,C=COMP.
REWIND,COMP.
CATALOG,COMP,COMP,ID=SLD,RP=1.
PURGE,PREPPL,ID=SLD.
7/8/9
        *INSERT ISELXK.100
        ISELXK=.FALSE.
        *DELETE MAINP.93
        *DELETE MAINP.186
        *DELETE MAINP.197
        *DELETE MAINP.207
        *DELETE PFILE.5
```

Step 6. Compile and load the PREPRO program.

```
ATTACH,COMP,ID=SLD.
REQUEST,PREPRO,PF.
```

```
FTN,I=COMP,L=0.
LOAD,LGO.
NOGO,PREPRO.
CATALOG,PREPRO,PREPRO,ID=SLD,RP=1.
PURGE,COMP,ID=SLD.
```

APPENDIX D

EXAMPLE INPUT FOR PROCESS

```
TITLE  PROJECT=EXAMPLE 6,USER=S DEROULHAC
COMPONENT DATA
  LIBID 1,H2/2,TOLUENE/3,BENZENE/4,,16020160/5,METHANE
THERMODYNAMIC DATA
  TYPE SYSTEM=CS
STREAM DATA
  PROP  STRM=1,TEMP=70,PRES=14.696,RATE=100,COMP=2,100
  PROP  STRM=17,TEMP=70,PRES=100,RATE=306,COMP=1,300/4,6
  PROP  STRM=8,TEMP=940,PRES=490,RATE=481,COMP=1,300/25/75/6/75
  PROP  STRM=16,TEMP=230,PRES=25,RATE=25.6,COMP=1/22.5/1/0.1/1
UNIT OPERATIONS
  MIXER  UID=1,NAME=M201
    FEED  1,16
    PROD  L=2
  PUMP  UID=2,NAME=P202
    FEED  2
    PROD  L=3
    OPER  POUT=500
  COMPRESSOR  UID=3,NAME=P206
    FEED  17
```

```
  PROD  V=4
  OPER  POUT=500
MIXER  UID=4,NAME=M203
  FEED  3,4
  PROD  M=5
HX  UID=5,NAME=E204
  HOT   FEED=8,M=9,DP=5
  COLD  FEED=5,M=6,DP=5
  SPEC  HOCI=20
  CONF  U=70
HX  UID=6,NAME=F205
  COLD  FEED=6,V=7,DP=5
  SPEC  COLD,TEMP=940
  UTILITY  STEAM,TSAT=1200
REACTOR  UID=7,NAME=R301
  FEED  7
  PROD  V=8
  OPERATION PHASE=V
  CALCULATION CONVERSION
  STOICHIOMETRIC 1,-1/-1/1/0/1
  BASE  COMP=2
  CONVERSION 0.75
FLASH  UID=8,NAME=D401
  FEED  9
  PROD  V=10,L=11
  ISO  TEMP=40,PRES=30
```

```
  HX  UID=9,NAME=E402
    COLD  FEED=11,M=12,DP=5
    SPECIFICATION  COLD,TEMP=210
    CONFIG U=15
    UTILITY  STEAM,TSAT=250
  SHORTCUT  UID=10,NAME=C403
    FEED  12
    PROD  STRM=13,PHASE=V
    PROD  STRM=14,PHASE=L
    CONDENSER TYPE=1
    CALCULATION KEYL=3,KEYH=2,RRMIN=3
    SPEC STRM=13,COMP=3,RATE=0.99,REFFEEDS
    SPEC STRM=13,COMP=2,RATE=0.008,REFFEEDS
  SPLITTER  UID=11,NAME=S501
    FEED  14
    PROD  M=15,M=16
    SPEC  STRM=15,RATE=0.1,RATIO,REFFEED
    SPEC  STRM=16,RATE=0.9,RATIO,REFFEED
  HX  UID=12,NAME=E601
    HOT  FEED=13,M=18,DP=5
    SPEC HOT,TEMP=70
    CONFIGURATION  U=90
    UTILITY  WATER,TIN=50,TOUT=65
RECYCLE DATA
  LOOP  NO=2,START=1,END=11,TOLERANCE=0.01,TRIAL=20
```

APPENDIX E

FLOWTRAN INSTALLATION

This Appendix contains the commands to install the FLOWTRAN program on a CDC CYBER 175 and on a VAX 11/780. The CYBER procedure is completed in steps.

Step 1. The FLOWTRAN program comes on a tape in internal format. The CYBER needs the tape to be in system internal, SI, format. These commands copy the internal format tape onto a second SI format tape. After this procedure is completed, the original tape is no longer needed. Only the copy is used, and it will subseqently be called the FLOWTRAN tape.

```
REQUEST,TAPE1,VSN=7430E,PE,S.
REQUEST,TAPE2,VSN=7478E,PE,RI=FLWTRN.
COPYNOS,TAPE1,TAPE2.
SCAN,TAPE2/R.
```

Step 2. Unload the files from the tape. The example problems (those files starting with DAT) are not stored on the computer. They are simply printed. The files RDFLOW, FLOWT, and FLOEXE are not stored either because they cannot be used with the NOS/BE operating system.

```
REQUEST,TAPE2,VSN=7478E,PE.
COPYBF,TAPE2,RDFLOW.
```

```
RETURN,RDFLOW.
COPYBF,TAPE2,FLOWT.
RETURN,FLOWT.
REQUEST,INF,PF.
COPYBF,TAPE2,INF.
CATALOG,INF,INF,ID=SLD,RP=1.
RETURN,INF.
REQUEST,PREPRO,PF.
COPYBF,TAPE2,PREPRO.
CATALOG,PREPRO,PREPRO,ID=SLD,RP=1.
RETURN,PREPRO.
REQUEST,VLE,PF.
COPYBF,TAPE2,VLE.
CATALOG,VLE,VLE,ID=SLD,RP=1.
RETURN,VLE.
REQUEST,PROPTY,PF.
COPYBF,TAPE2,PROPTY.
CATALOG,PROPTY,PROPTY,ID=SLD,RP=1.
RETURN,PROPTY.
REQUEST,FTSLIB,PF.
COPYBF,TAPE2,FTSLIB.
CATALOG,FTSLIB,FTSLIB,ID=SLD,RP=1.
RETURN,FTSLIB.
REQUEST,FTPUBF,PF.
COPYBF,TAPE2,FTPUBF.
CATALOG,FTPUBF,FTPUBF,ID=SLD,RP=1.
```

```
RETURN,FTPUBF.
REQUEST,FTPRIF,PF.
COPYBF,TAPE2,FTPRIF.
CATALOG,FTPRIF,FTPRIF,ID=SLD,RP=1.
RETURN,FTPRIF.
COPYBF,TAPE2,FLOEXE.
RETURN,FLOEXE.
COPYBF,TAPE2,FT1.
REWIND,FT1.
COPYSBF,FT1,OUTPUT.
RETURN,FT1.
COPYBF,TAPE2,INF51.
REWIND,INF51.
COPYSBF,INF51,OUTPUT.
RETURN,INF51.
COPYBF,TAPE2,VL31.
REWIND,VL31.
COPYSBF,VL31,OUTPUT.
RETURN,VL31.
COPYBF,TAPE2,PR41.
REWIND,PR41.
COPYSBF,PR41,OUTPUT.
RETURN,PR41.
REQUEST,FTPID,PF.
COPYBF,TAPE2,FTPID.
```

```
CATALOG,FTPID,FTPID,ID=SLD,RP=1.
RETURN,FTPID.
REQUEST,FTBTF,PF.
COPYBF,TAPE2,FTBTF.
CATALOG,FTBTF,FTBTF,ID=SLD,RP=1.
RETURN,FTBTF.REQUEST,DAT1,PF.
COPYBF,TAPE2,DAT1.
REWIND,DAT1.
COPYSBF,DAT1,OUTPUT.
RETURN,DAT1.
COPYBF,TAPE2,DAT2.
REWIND,DAT2.
COPYSBF,DAT2,OUTPUT.
RETURN,DAT2.
```

The last four lines can be repeated for a printed copy of the other data and output files.

Step 3. Change the library, FTSLIB, from NOS to NOS/BE.

```
REQUEST,FTSLIB,PF.
ATTACH,X,FTSLIB,ID=SLD.
COPYBR,X,DUMMY,1.
COPYBR,X,Y,179.
RETURN,X.
EDITLIB,USER.
CATALOG,FTSLIB,ID=SLD,RP=1.
7/8/9
      LIBRARY(FTSLIB,NEW)
```

```
ADD(*,Y)
FINISH.
ENDRUN.
```

Step 4. Input the new FLOEXE commands.

```
REQUEST,FLOEXE,PF.
COPYBF,INPUT,FLOEXE.
CATALOG,FLOEXE,ID=SLD,RP=1.
7/8/9
      .PROC,FT,IN,OUT.
      REQUEST,OUT,PF.
      ATTACH,TAPE5,IN,ID=SLD.
      ATTACH,TAPE18,FTPRIF,ID=SLD.
      ATTACH,TAPE19,FTPUBF,ID=SLD.
      ATTACH,FTSLIB,ID=SLD.
      ATTACH,INF,ID=SLD.
      LDSET,LIB=FTSLIB,PRESET=ZERO.
      INF.
      RETURN,TAPE5.
      SKIPEI,TAPE6.
      REWIND,TAPE4.
      COPYBF,TAPE4,TAPE5.
      REWIND,TAPE5.
      ATTACH,TAPE8,FTBTF,ID=SLD.
      ATTACH,PREPRO,ID=SLD.
      REWIND,FTSLIB.
```

```
LDSET,LIB=FTSLIB,PRESET=ZERO.
PREPRO.
RETURN,TAPE5,TAPE1.
REWIND,TAPE11.
COPYBF,TAPE11,TAPE5.
REWIND,TAPE5.
SKIPEI,TAPE6.
REWIND,TAPE9.
FTN5,I=TAPE9.
LDSET,LIB=FTSLIB,USEP=INPUTD,PRESET=ZERO.
LGO.
CATALOG,TAPE6,OUT,ID=SLD,RP=1.
REVERT.
(control z)
.PROC,FTBT.
REQUEST,FTBTD,PF.
REQUEST,FTBTF,PF.
ATTACH,FTBT,ID=SLD.
ATTACH,TAPE5,FTBTD,ID=SLD.
FTN5,I=FTBT,L=0.
LGO.
CATALOG,TAPE6,FTBTD,ID=SLD,RP=1.
CATALOG,TAPE8,FTBTF,ID=SLD,RP=1.
REVERT.
(control z)
.PROC,FTPRI.
```

```
ATTACH,FTPRI,ID=SLD.
ATTACH,TAPE18,FTPRIF,ID=SLD.
FTN5,I=FTPRI,L=0.
LGO.
REVERT.
(control z)
.PROC,FTPUB.
ATTACH,FTPUB,ID=SLD.
ATTACH,TAPE19,FTPUBF,ID=SLD.
FTN5,I=FTPUB,L=0.
LGO.
REVERT.
(control z)
.PROC,INF,IN,OUT.
REQUEST,OUT,PF.
ATTACH,TAPE5,IN.
ATTACH,TAPE4,FTPID.
ATTACH,TAPE18,FTPRIF,ID=SLD.
ATTACH,TAPE19,FTPUBF,ID=SLD.
ATTACH,FTSLIB,ID=SLD.
ATTACH,INF,ID=SLD.
ATTACH,INF1,ID=SLD.
LDSET,LIB=FTSLIB,PRESET=ZERO.
INF1.
CATALOG,TAPE6,OUT,RP=1,ID=SLD.
```

```
REVERT.
(control z)
.PROC,PROPTY,IN,OUT.
REQUEST,OUT,PF.
ATTACH,TAPE5,IN,ID=SLD.
ATTACH,FTSLIB,ID=SLD.
ATTACH,PROPTY,ID=SLD.
LDSET,LIB=FTSLIB,PRESET=ZERO.
PROPTY.
RETURN,TAPE5.
COPY,TAPE7,TAPE5.
REWIND,TAPE5.
SKIPEI,TAPE6.
ATTACH,TAPE4,FTPID.
ATTACH,TAPE18,FTPRIF,ID=SLD.
ATTACH,TAPE19,FTPUBF,ID=SLD.
ATTACH,INF,ID=SLD.
LDSET,LIB=FTSLIB,PRESET=ZERO.
INF.
CATALOG,TAPE6,OUT,ID=SLD,RP=1.
REVERT.
(control z)
.PROC,VLE,IN,OUT.
REQUEST,OUT,PF.
ATTACH,TAPE5,IN,ID=SLD.
ATTACH,TAPE4,FTPID,ID=SLD.
```

```
ATTACH,TAPE18,FTPRIF,ID=SLD.
ATTACH,TAPE19,FTPUBF,ID=SLD.
ATTACH,FTSLIB,ID=SLD.
ATTACH,INF,ID=SLD.
LDSET,LIB=FTSLIB,PRESET=ZERO.
INF.
RETURN,TAPE5.
REWIND,TAPE4.
COPYBF,TAPE4,TAPE5.
REWIND,TAPE5.
SKIPEI,TAPE6.
LDSET,LIB=FTPLIB/FTSLIB,USEP=VLEBLK,PRESET=ZERO.
ATTACH,VLE,ID=SLD.
LDSET,LIB=FTSLIB,PRESET=ZERO.
VLE.
CATALOG,TAPE6,OUT,ID=SLD,RP=1.
REVERT.
EXIT.
```

The program is now ready to execute. To execute the program, the commands below need to be given. If the VLE, INF, or PROPTY programs are desired, replace FT with their name.

```
ATTACH,FLOEXE,ID=SLD.
BEGIN,FT,FLOEXE,INPUT,OUTPUT.
7/8/9
```

input data

Installing FLOWTRAN on the VAX can be done in one step.

```
$ ALLOCATE MF:
$ REQUEST/REPLY "PLEASE MOUNT TAPE #3141F"
$ MOUNT/FOREIGN MF:FLOTRN
$ BACKUP MF:FLOTRN *.*
$ DISMOUNT MF:
$ DEALLOCATE MF:
```

To make a copy of the FLOWTRAN tape, unload the files into a directory, and then save the files to a new tape.

```
$ ALLOCATE MF:
$ REQUEST/REPLY "PLEASE MOUNT TAPE #xxxx"
$ INITIALIZE MF:FLOTRN
$ MOUNT/FOREIGN MF: FLOTRN
$ BACKUP/VERIFY *.* MF:FLOTRN
$ DISMOUNT MF:
$ DEALLOCATE MF:
```

Note: If the VAX 2 or VAX 1 is used, the device driver is named MT and not MF.

The execution on the VAX is just as easy. Simply place the input data in a file, and type the appropriate command listed below. Similar to the CYBER, to execute INF, VLE, or PROPTY, replace FT with the program name.

```
@FT file name
```

APPENDIX F

FLOWTRAN EXAMPLE

A. Input Data

```
TITLE TOLUENE DEALKYLATION
PROPS 5 2 1 1 2
PRINT TABLES
RETR HYDROGEN TOLUENE BENZENE NITROGEN C1
BLOCK M201 ADD S1 S16B 5*0 S2
BLOCK P202 PUMP S2 S3
PARAM P202 1 500
BLOCK P206 GCOMP S17 S4
PARAM P206 1 500 0 4 0 0 1
BLOCK M203 ADD S3 S4 5*0 S5
PARAM M203 1 0 0
BLOCK E204 EXCH3 S8 S5 S9 S6
PARAM E204 1 20 5 5 70 1 1 0
BLOCK F205 HEATR S6 S7
PARAM F205 1 1040 5 0 0 1
BLOCK R301 REACT S7 6*0 S8 0
PARAM R301 1 1040 490 0 1 2 0.75 -1 -1 1 0 1
BLOCK D401 IFLSH S9 S11 S10
PARAM D401 1 40 30 0
BLOCK E402 HEATR S11 S12
```

```
PARAM E402 1 210 5 0 0 1 0
BLOCK C403 DSTWU S12 S14 S13
PARAM C403 1 2 3 99 124 0 -3 0 25 25 1
BLOCK S501 SPLIT S14 S15 S16A 5*0
PARAM S501 1 2 0.1 0.9
BLOCK CON1 SCVW S16A 2*0 M201 S16B 2*0
PARAM CON1 2 0.01 20 1
BLOCK E601 HEATR S13 S18
PARAM E601 1 70 5 0 0 1
MOLES S1 2 100
TEMP S1 70
PRESS S1 14.696
TEMP S17 70
PRESS S17 100
MOLES S17 1 300 0 0 6
TEMP S16B 230
PRESS S16B 25
MOLES S16B 1 22.5 1 0.1 1
TEMP S8 1040
PRESS S8 490
MOLES S8 1 300 25 75 6 75
END CASE
END JOB
```

B. Output Results

UNIT TABLE

UNIT NAME	UNIT TYPE
M201	ADD
P202	PUMP
P206	GCOMP
M203	ADD
E204	EXCH3
F205	HEATR
R301	REACT
D401	IFLSH
E402	HEATR
C403	DSTWU
S501	SPLIT
CON1	SCVW
E601	HEATR

FLOW TABLE

STREAM NAME	FROM	TO
S1		M201
S10	D401	
S11	D401	E402
S12	E402	C403
S13	C403	E601
S14	C403	S501
S15	S501	
S16A	S501	CON1
S16B	CON1	M201
S17		P206
S18	E601	
S2	M201	P202
S3	P202	M203
S4	P206	M203
S5	M203	E204
S6	E204	F205
S7	F205	R301
S8	R301	E204
S9	E204	D401

INPUT STREAMS
S1
S17

OUTPUT STREAMS
S10
S15
S18

14-FEB-8

TOLUENE DEALKYLATION

PHYSICAL PROPERTY OPTIONS
CAVETT VAPOR PRESSURE
IDEAL GAS VAPOR FUGACITY
UNCORRECTED LIQUID FUGACITY
SCATCHARD-HILDEBRAND ACTIVITY COEF

14-FEB-8

TOLUENE DEALKYLATION

M201 (ADD) T=114.93 F, P=14.70 PSIA, V/F=0.0000 MOLS/MOL
FEEDS = S1 S16B
PRODUCT = S2

P202 - PUMP - INLET = S2, OUTLET = S3
FLOW,GPM = 27.59, DELTA P,PSI = 485.30
FLUID HP = 7.81, BRAKE HP = 22.41, ELEC KW = 18.92
PUMP EFF = 0.2486, DRIVER EFF = 0.8830

P206 (GCOMP) INLET = S17 OUTLET = S4
OUTLET PRESSURE, PSIA=500.00 OUTLET TEMP, DEG F=378.33
ISENTROPIC TEMP, DEG F = 378.33
ISENTROPIC HORSEPOWER = 261.2
INDICATED HORSEPOWER = 261.2 BRAKE HORSEPOWER = 261.2

M203 (ADD) T=185.47 F, P=500.00 PSIA, V/F=0.7129 MOLS/MOL
FEEDS = S3 S4
PRODUCT = S5

E204 - EXCH3 - HEAT EXCHANGED = 0.7678E+07 BTU/HR
CORRECTED LMTD (DEG F) = 48.25, OVERALL U = 70.0
CALCULATED AREA (SQ FT) = 2273.37

INLETS =	S8	S5	OUTLETS =	S9	S6
DEG F	1040.00	185.47		205.47	944.64

F205 - HEATR - INLET = S6, OUTLET = S7
OUTLET TEMP = 1040.00 DEG F, PRESSURE DROP = 5.00 PSI
DUTY = 0.8969E+06 BTU/HR

R301 - REACTOR BLOCK
TEMP= 1040.00 DEG F PRESS= 490.00 PSIA
INPUT STREAMS- S7
OUTPUT STREAM- S8

D401 (IFLSH) T=40.00 F, P=30.00PSIA, V/F=0.7273 MOLS/MOL
FEED = S9, BOTTOMS = S11, OVERHEAD = S10
HEAT ADDED = -0.1131E+07 BTU/HR

E402 - HEATR - INLET = S11, OUTLET = S12
OUTLET TEMP = 210.00 DEG F, PRESSURE DROP = 5.00 PSI
DUTY = 0.6842E+06 BTU/HR

C403 (DSTWU) FEED=S12 OVD=S13 BOT=S14

HVY KEY COMP. NO.	2.000	LT. KEY COMP. NO.	3.000
SPLIT FOR LT. KEY	99.000	SPLIT FOR HVY KEY	124.000
QUALITY OF FEED	0.000		
DESIRED REFLUX	-3.000	DESIRED NO OF STGS	0.000
TOP PRESS. PSIA	25.000	BOTTOM PRESS. PSIA	25.000

```
     CONDENSER TYPE                              1.000
     MIN. NO. THEO. STAGES AT TOTAL REFLUX      11.32
     MINIMUM REFLUX AT INFINITE STAGES           0.97
     NO. STAGES ABOVE FEED AT TOTAL REFLUX       4.33
     ACTUAL REFLUX                               2.91
     NO. OF THEO. STAGES AT ACTUAL REFLUX       14.57
     CONDENSER TEMP, DEG F                     210.18
     REBOILER TEMP, DEG F                      265.73

S501 - SPLIT - INPUT = S14
     OUTLETS =    S15      S16A
     FRACTION =  0.1000   0.9000

E601 - HEATR - INLET = S13, OUTLET = S18
     OUTLET TEMP =   70.00 DEG F, PRESSURE DROP =  5.00 PSI
     DUTY = -0.1515E+07 BTU/HR
```

14-FEB-8

TOLUENE DEALKYLATION

STREAM NAME:	S1	S10	S11	S12
	LBMOL/HR	LBMOL/HR	LBMOL/HR	LBMOL/HR
1 HYDROGEN	0.0	208.632	0.04191	0.04191
2 TOLUENE	100.000	0.46012	29.9821	29.9821
3 BENZENE	0.0	5.02887	86.5175	86.5175
4 NITROGEN	0.0	5.99674	0.00326	0.00326
5 METHANE	0.0	91.1429	0.18361	0.18361
TOTAL LBMOL/HR	100.000	311.260	116.728	116.728
TOTAL LB/HR	9213.40	2485.91	9523.20	9523.20
1000 BTU/HR	-1478.51	90.86	-1694.05	-1009.81
DEGREES F	70.00	40.00	40.00	210.00
PSIA	14.696	30.000	30.000	25.000
DENSITY, LB/FT3	54.1401	0.0447	55.5739	0.0000
MOLE FRAC VAPOR	0.0000	1.0000	0.0000	0.0108

STREAM NAME:	S13	S14	S15	S16A
	LBMOL/HR	LBMOL/HR	LBMOL/HR	LBMOL/HR
1 HYDROGEN	0.04191	0.00000	0.00000	0.00000
2 TOLUENE	0.69577	29.2863	2.92863	26.3577
3 BENZENE	86.2220	0.29555	0.02955	0.26599
4 NITROGEN	0.00326	0.00000	0.00000	0.00000
5 METHANE	0.18361	0.00000	0.00000	0.00000
TOTAL LBMOL/HR	87.1465	29.5818	2.95818	26.6237
TOTAL LB/HR	6801.85	2721.35	272.135	2449.21
1000 BTU/HR	366.06	-201.52	-20.15	-180.84
DEGREES F	210.18	265.73	265.73	265.73
PSIA	25.000	25.000	25.000	25.000
DENSITY, LB/FT3	0.2824	47.4052	47.4052	47.4052
MOLE FRAC VAPOR	1.0000	0.0000	0.0000	0.0000

STREAM NAME:	S16B	S17	S18	S2
	LBMOL/HR	LBMOL/HR	LBMOL/HR	LBMOL/HR
1 HYDROGEN	0.0	300.000	0.04191	0.0
2 TOLUENE	26.2936	0.0	0.69577	126.294
3 BENZENE	0.26535	0.0	86.2220	0.26535
4 NITROGEN	0.0	6.00000	0.00326	0.0
5 METHANE	0.0	0.0	0.18361	0.0
TOTAL LBMOL/HR	26.5590	306.000	87.1465	126.559
TOTAL LB/HR	2443.26	772.896	6801.85	11656.7
1000 BTU/HR	-180.84	145.02	-1148.83	-1659.35
DEGREES F	265.73	70.00	70.00	114.93
PSIA	25.000	100.000	20.000	14.696
DENSITY, LB/FT3	0.0000	0.0443	0.0000	52.6790
MOLE FRAC VAPOR	0.0002	1.0000	0.0009	0.0000

STREAM NAME:	S3	S4	S5	S6
	LBMOL/HR	LBMOL/HR	LBMOL/HR	LBMOL/HR
1 HYDROGEN	0.0	300.000	300.000	300.000
2 TOLUENE	126.294	0.0	126.294	126.294
3 BENZENE	0.26535	0.0	.26535	0.26535
4 NITROGEN	0.0	6.00000	6.00000	6.00000
5 METHANE	0.0	0.0	0.0	0.0
TOTAL LBMOL/HR	126.559	306.000	432.559	432.559
TOTAL LB/HR	11656.7	772.896	12429.6	12429.6
1000 BTU/HR	-1659.35	809.54	-849.81	6828.10
DEGREES F	114.93	378.33	185.47	944.64
PSIA	500.000	500.000	500.000	495.000
DENSITY, LB/FT3	52.6790	0.1384	0.0000	0.9396
MOLE FRAC VAPOR	0.000	1.0000	0.712	1.0000

STREAM NAME:	S7	S8	S9
	LBMOL/HR	LBMOL/HR	LBMOL/HR
1 HYDROGEN	300.000	205.280	208.673
2 TOLUENE	126.294	31.5734	30.4422
3 BENZENE	0.26535	94.9856	91.3266
4 NITROGEN	6.00000	6.00000	6.00000
5 METHANE	0.0	94.7202	91.3266
TOTAL LBMOL/HR	432.559	432.559	427.989
TOTAL LB/HR	12429.6	12429.6	12009.1
1000 BTU/HR	7725.04	7390.91	-471.85
DEGREES F	1040.00	1040.00	205.47
PSIA	490.000	490.000	485.000
DENSITY, LB/FT3	0.8699	0.8716	0.0000
MOLE FRAC VAPOR	1.0000	1.0000	0.7374

```
                                                    14-FEB-8
TOLUENE DEALKYLATION

  **       14-FEB-8    13:38:35
M201 (ADD) T= 78.64 F, P=14.70 PSIA, V/F=0.1960
 ***P202  FEED NOT ALL LIQ., OUTLET T AND FV MAY BE WRONG
P202 - OUTLET PRES= 500.0PSIA, FLOW=  124.6LB-MOL/HR
P206 (GCOMP)  IN  T =  70.00 F,  P = 100.00 PSIA
              OUT T = 378.33 F,  P = 500.00 PSIA
M203 (ADD) T=175.89 F, P=500.00 PSIA, V/F=0.7706
E204 - Q= 0.709E+07 OUTLET T = 195.9  998.9
F205 - TEMP = 1040.00F, DELP = -5.0PSI, Q= 0.337E+06BTU/HR
R301 - REACTOR BLOCK
D401 (IFLSH) T = 40.00 F, P = 30.00PSIA, V/F = 0.8060
             Q = -0.102E+07 BTU/HR
E402 - TEMP = 210.0F, DELP= -5.0PSI, Q= 0.544E+06BTU/HR
C403 (DSTWU)  FEED=S12  OVD=S13  BOT=S14
             ITR TOP TEMP BOT TEMP MIN THEO STGS
               1   209.073   239.073        11.333
               2   210.219   265.730        11.327
             MINIMUM REFLUX    0.981
S501 - SPLITS S14  INTO 2 OUTLETS
   *CON1  -SCVW-ITER=  1, TIME=  0SEC, MAX ERROR=20.47726
     S16A(NEW)  S16B(OLD)   ERROR
   1-0.359E-07   22.5     -1.000*
   2   21.5       1.00     20.477*
   3  0.217      0.100      0.117*
   4-0.194E-08   1.00     -1.000*
   5-0.899E-07 0.000E+00  0.000
   6   21.7       24.6     -0.118
   7   266.       230.      0.155*
   8   25.0       25.0      0.000
   9-0.148E+06 0.437E+05-4.382
  10  0.000E+00   1.00     -1.000
M201 (ADD) T= 108.31 F, P=14.70 PSIA, V/F=0.0000
P202 - OUTLET PRES= 500.0PSIA, FLOW=  121.7LB-MOL/HR
P206 (GCOMP)  IN  T =  70.00 F,  P = 100.00 PSIA
              OUT T = 378.33 F,  P = 500.00 PSIA
M203 (ADD) T=183.41 F, P=500.00 PSIA, V/F=0.7208
E204 - Q= 0.679E+07 OUTLET T = 203.4  868.9
F205 - TEMP = 1040.00F, DELP = -5.0PSI, Q= 0.154E+07BTU/HR
R301 - REACTOR BLOCK
D401 (IFLSH) T = 40.00 F, P = 30.00PSIA, V/F = 0.7786
             Q = -0.103E+07 BTU/HR
E402 - TEMP = 210.0F, DELP= -5.0PSI, Q= 0.557E+06BTU/HR
C403 (DSTWU)  FEED=S12  OVD=S13  BOT=S14
             ITR TOP TEMP BOT TEMP MIN THEO STGS
               1   210.219   265.730        11.321
             MINIMUM REFLUX    0.976
S501 - SPLITS S14  INTO 2 OUTLETS
   *CON1  -SCVW-ITER=  2, TIME=  0SEC, MAX ERROR= 0.01338
```

```
     S16A(NEW)  S16B(OLD)   ERROR
   1-0.348E-07  0.000E+00 0.000
   2   21.8        21.5      0.013*
   3  0.220       0.217      0.003
   4-0.267E-08  0.000E+00 0.000
   5-0.107E-06  0.000E+00 0.000
   6   22.0        21.7      0.013*
   7   266.        266.      0.000
   8   25.0        25.0      0.000
   9-0.150E+06-0.148E+06 0.014
  10  0.000E+00  0.221E-03  0.000
M201 (ADD) T= 108.74 F, P=14.70 PSIA, V/F=0.0000
P202 - OUTLET PRES= 500.0PSIA, FLOW=  122.0LB-MOL/HR
P206 (GCOMP)  IN  T =  70.00 F,  P = 100.00 PSIA
              OUT T = 378.33 F,  P = 500.00 PSIA
M203 (ADD) T=183.54 F, P=500.00 PSIA, V/F=0.7203
E204 - Q= 0.768E+07 OUTLET T = 203.5  967.5
F205 - TEMP = 1040.00F, DELP = -5.0PSI, Q= 0.665E+06BTU/HR
R301 - REACTOR BLOCK
D401 (IFLSH) T = 40.00 F, P = 30.00PSIA, V/F = 0.7278
             Q = -0.111E+07 BTU/HR
E402 - TEMP = 210.0F, DELP= -5.0PSI, Q= 0.682E+06BTU/HR
C403 (DSTWU)  FEED=S12  OVD=S13  BOT=S14
             ITR TOP TEMP BOT TEMP MIN THEO STGS
               1   210.206   265.731        11.321
             MINIMUM REFLUX    0.971
S501 - SPLITS S14  INTO 2 OUTLETS
   *CON1  -SCVW-ITER=  3, TIME=  0SEC, MAX ERROR= 0.20786
     S16A(NEW)  S16B(OLD)   ERROR
   1-0.386E-07  0.000E+00 0.000
   2   26.3        21.8      0.208*
   3  0.265       0.220      0.046*
   4-0.300E-08  0.000E+00 0.000
   5-0.168E-06  0.000E+00 0.000
   6   26.6        22.0      0.208*
   7   266.        266.      0.000
   8   25.0        25.0      0.000
   9-0.181E+06-0.1450+06 0.209
  10  0.000E+00  0.342E-03  0.000
M201 (ADD) T= 114.93 F, P=14.70 PSIA, V/F=0.0000
P202 - OUTLET PRES= 500.0PSIA, FLOW=  126.6LB-MOL/HR
P206 (GCOMP)  IN  T =  70.00 F,  P = 100.00 PSIA
              OUT T = 378.33 F,  P = 500.00 PSIA
M203 (ADD) T=185.47 F, P=500.00 PSIA, V/F=0.7129
E204 - Q= 0.768E+07 OUTLET T = 205.5  944.6
F205 - TEMP = 1040.00F, DELP = -5.0PSI, Q= 0.897E+06BTU/HR
R301 - REACTOR BLOCK
D401 (IFLSH) T = 40.00 F, P = 30.00PSIA, V/F = 0.7273
             Q = -0.113E+07 BTU/HR
E402 - TEMP = 210.0F, DELP= -5.0PSI, Q= 0.684E+06BTU/HR
C403 (DSTWU)  FEED=S12  OVD=S13  BOT=S14
```

```
            ITR TOP TEMP BOT TEMP MIN THEO STGS
              1   210.178   265.731        11.321
            MINIMUM REFLUX     0.971
S501 - SPLITS S14   INTO 2 OUTLETS
   *CON1  -SCVW CONVERGED-I=  4, TIME=  0SEC, MAX ERROR=
0.00243
     S16A(NEW)  S16B(OLD)   ERROR
   1-0.386E-07 0.000E+00 0.000
   2   26.4       26.3      0.002
   3  0.266      0.265      0.001
   4-0.301E-08 0.000E+00 0.000
   5-0.169E-06 0.000E+00 0.000
   6   26.6       26.6      0.002
   7   266.       266.      0.000
   8   25.0       25.0      0.000
   9-0.181E+06-0.1810+06 0.003
  10  0.000E+00  0.246E-03 0.000
E601 - TEMP =   70.0F, DELP = -5.0PSI,  Q=-0.151E+07BTU/HR
  **END OF HISTORY
```

LIST OF REFERENCES

Brannock, N. F., V. S. Verneuil, and Y. L. Wang. "PROCESS Simulation Program," Computer-Aided Process Plant Design, Gulf Publishing Company, Houston, 1982, p. 536 - 545.

Brannock, N. F., V. S. Verneuil, and Y. L. Wang. "PROCESS Simulation Program: A Comprehensive Flowsheeting Tool for Chemical Engineers," Computers and Chemical Engineering, Vol. 3 (1979), p. 329 - 352.

Briddell, E. T. "Process Design by Computer," Chemical Engineering, Vol. 83, No. 3 (1974), p. 60, No. 5, p. 113, No. 7, p. 77.

Evans, L. B. "Advances in Process Flowsheeting Systems," Foundations of Computer-Aided Chemical Process Design, Vol. 1 (1981), p. 425 - 469.

Fair, J. R. "Advanced Process Engineering", A.I.Ch.E. Monograph Series, Vol. 76, No. 13 (1980),

Flower, J. R., and B. D. Whitehead. "Computer-aided Design: A Survey of Flowsheeting Programs," The Chemical Engineer, No. 272 (April 1973), p. 208 - 210, No. 273 (May 1973), p. 271 - 277.

Harris, R. E. "Distillation Designs Using FLOWTRAN," Chemical Engineering Progress, Vol. 68, No. 10 (Oct. 1972), p. 56 - 61.

King, C. J., A. S. Foss, E. A. Grens, S. Lynn, and D. F. Rudd. "Chemical Process Design and Engineering," Chemical Engineering Education, Vol. 7, No. 2 (1973), p. 72 - 74.

Kirk-Othmer Encyclopedia of Chemical Technology, 2nd. Edition, USA: John Wiley and Sons, Inc., 1964, Vol. 3, p. 379.

Larsen, Alvin H. "FLOWTRAN for Process Simulation," Computer Aided Process Plant Design, Gulf Publishing Company, Houston, 1982, p. 475 - 488.

Leesley, Michael. "Evaluating Process Simulation Software," Computer-Aided Process Plant Design, Gulf Publishing Company, Houston, 1982, p. 131 - 141.

Massey, Nathan A. "The Process Design Office," Computer-Aided Process Plant Design, Gulf Publishing Company, Houston, 1982, p. 190 - 230.

Mattione, M. J., W. J. Meier, and N. L. Book. "An Equation Oriented Approach to the Structuring and Solution of Chemical Process Design Problems," Selected Topics of Computer-Aided Process Design and Analysis A. I. Ch. E. Symposium Series No. 214, Vol. 78 (1982), p. 1 - 37.

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

Motard, R. L., and H. M. Lee. CHESS User's Guide, University of Houston Publishing, Houston, 1971.

Motard, R. L., M. Shacham, and E. M. Rosen. "Steady State Chemical Process Simulation," A. I. Ch. E. Journal, Vol. 21, No. 3 (May 1975), p. 417 - 436.

Murtagh, B. A. "On the Simultaneous Solution and Optimization of Large-Scale Engineering Systems," Computers and Chemical Engineering, Vol. 6 (1982).

Orbach, O., and C. M. Crowe. "Convergence Promotion in the Computing of Chemical Processes with Recycle - The Dominant Eigenvalue Method," Canadian Journal of Chemical Engineering, Vol. 49, No. 4 (1971), p. 509.

Peters, N., and P. E. Barker. "An Appraisal of the Use of PACER, GEMCS, and CONCEPT for Chemical Plant Simulation and Design," The Chemical Engineer, No. 283 (March 1974), p. 149 - 157.

Peterson, Jeffery N., Chau-Chyun Chen, and Lawrence B. Evans. "Computer Programs for Chemical Engineers: 1978 - " Chemical Engineering, Vol. 85, No. 13 (June 5, 1978), p. 145-52, 154.

PROCESS 0883 Installation Guide for CDC NOS Systems, Simulation Sciences, Inc., Fullerton, California, 1984.

PROCESS Simulation Program Input Manual, Simulation Sciences, Inc., Fullerton, California, 1983.

PROCESS Simulation Key Word Input Guide, Simulation Sciences, Inc., Fullerton, California, 1983.

Proctor, Stanley I. "The FLOWTRAN Simulation System," Chemical Engineering Progress, Vol. 79, No. 6 (June 1983), p. 49 - 53.

Reid, Robert C., John M. Prausnitz, and Thomas K. Sherwood. The Properties of Gases and Liquids, 3rd. Edition, McGraw-Hill, New York, 1977.

Rorschach, Robert L., and Robert E. Harris. "Process Simulation Made by Computer," Oil and Gas Journal, Vol. 68, No. 33 (Aug. 17, 1970), p. 62 - 66.

Rose, L. M. The Application of Mathematical Modelling to Process Development and Design, Applied Science Publishers, LTD., London, 1974.

Rosen E. M., and D. J. Kaufman. "Computers, Design and Monsanto," Computers in Industry, Vol. 2 (1981), p. 41 - 57.

Rosen E. M., and A. C. Pauls. "Computer Aided Chemical Process Design: The FLOWTRAN System," Computers and Chemical Engineering, Vol. 1 (1977), p. 11-21.

Seader, J. D., W. D. Seider, and Allen C. Pauls. FLOWTRAN: An Introduction, CACHE, Massachusetts, 1977.

Seider, W. D. "Computer-aided Analysis and Design Packages," Chemical Engineering Computing, Vol. 2, p. 32.

Shacham, M., et. al. "Review: Equation Oriented Approach to Process Flowsheeting," Computers and Chemical Engineering, Vol. 6, No. 2 (1982), p. 79.

Umeda, T., and M. Nishio. "Comparison Between Sequential and Simulataneous Approaches in Process Simulation," Industrial and Engineering Chemistry: Process Design and Development, Vol. 11, No. 2 (1972), p. 153.

Westerberg, A. W., H. P. Hutchison, R. L. Motard, and P. Winter. Process Flowsheeting, Cambridge University Press, Cambridge, England, 1979.

Woods, D. R., et. al. "Teaching Experience with Design and Simulation Projects," Chemical Engineering Education, Vol. 7, No. 2 (1973), p. 96.

Worley, F. L. Jr., and R. L. Motard. "Information Systems in Chemical Engineering Design," Chemical Engineering Computing, Vol. 2, p. 4 - 24.