SYNTHESIS OF OPTIMAL ENERGY EXCHANGE NETWORKS USING DISCRETE METHODS

SYNTHESIS OF OPTIMAL ENERGY EXCHANGE NETWORKS USING DISCRETE METHODS

By

MICHAEL AYLMER MENZIES, B.E., M.Eng.

A Thesis

Submitted to the Faculty of Graduate Studies

in Partial Fulfilment of the Requirements

for the Degree

Doctor of Philosophy

McMaster University

January, 1972

DOCTOR OF PHILOSOPHY (1972)
(Chemical Engineering)

McMASTER UNIVERSITY
Hamilton, Ontario

TITLE: Synthesis of Optimal Energy Exchange Networks Using Discrete Methods

AUTHOR: Michael Aylmer Menzies, B.E. (University of Canterbury, N. Z.),
M.Eng. (McMaster University)

SUPERVISOR: Professor A. I. Johnson

NUMBER OF PAGES: (viii), 257.

SCOPE AND CONTENTS:

A flexible, modular program system for the synthesis of optimal energy exchange networks (OPENS) is developed. It is capable of generating realistic process equipment networks to satisfy both stream temperature and pressure specifications. The system contains elements of heuristic decision making and employs a "branch and bound" combinatorial technique for solving the discrete problem of optimizing network configuration. An (energy) price-based decomposition algorithm is developed for sub-process integration; this is achieved by determination of the optimal (stream) interconnections between such sub-processes.

The system is applied to the design of energy recovery networks for two quite dissimilar ethylene recovery schemes; the high and low pressure processes. Process interactions between the main processing sequence and the associated refrigeration facility are used to explore sub-process integration.

Some conclusions are made regarding the effectiveness of the program system for the example processes presented and recommendations are made for improvement and extensions.

ACKNOWLEDGEMENTS

The author is much indebted to his research director, Professor A. I. Johnson who provided valuable guidance, encouragement and interest during the course of this study.

Thanks are also due to the two other committee members, Professor C. M. Crowe and Professor J. W. Bandler, for their comments and suggestions.

The cooperation of the Dow Company of Canada in providing process data as well as practical advice on results obtained is gratefully acknowledged.

Financial assistance provided by the National Research Council of Canada and McMaster University was gratefully received.

TABLE OF CONTENTS

APPENDICES

LIST OF FIGURES

LIST OF TABLES

PART I

THEORY AND PROGRAM SYSTEM

CHAPTER I

INTRODUCTION

1.1 General

In recent years chemical process design has become increasingly automated. The design of many equipment units is now computerized and modular simulation systems are widely used for generalized mass and heat balancing and equipment sizing and costing for large process networks. Later advances have produced capabilities for automated process optimization and simulation of process dynamics. These simulation systems are in general capable only of analyzing a user-supplied flowsheet, leaving the creative aspects involved in process invention and evolution largely to the skill and ingenuity of the design engineer. However there is a growing interest in developing techniques for process synthesis, which is concerned with the analysis, ordering and automation of the logic required for process design decision making. Synthesis covers a broad and largely unexplored range from the evolution of a basic processing concept to the actual selection and arrangement of process equipment.

This study is concerned with the latter stages of the synthesis procedure. It reports on the development and application of OPENS (Optimal Process Equipment Network Synthesizer), a modularly oriented program system for the synthesis of optimal energy exchange networks. It combines recently developed theoretical concepts with practical design considerations to form a flexible system capable of generating very realistic, useful process designs.

1.2 Background

There are three general areas of the literature that form a background to this study. They are simulation, synthesis and optimization, and this section covers the relevant published work in each of these fields.

1.2.1. Simulation

The modular approach to steady state chemical process simulation[1] is now widely accepted. The basic concept is that of transforming the conventional process flowsheet into an information flow diagram in which process equipment are represented by closely corresponding computation modules. Computation of any process proceeds by sequential calculation of the individual module routines, a scheme which may need to be repeated if recycles are present. Manipulation of stream, equipment and other necessary information is handled by the simulation executive. The modular approach has the distinct advantages of this close and easily understood correspondence between process flowsheet and information flow diagram and a ready facility for altering process configurations. Further, within the modular approach, any number of equipment units of the same type may be represented by a single module with different parameter sets. An equation oriented approach to simulation[2] can also be used and such systems which are based on equation structure rather than plant structure may be computationally more efficient. However theoretical difficulties in solving large sets of generally non-linear equations and lack of convenience when compared with modular systems have prevented wide use of such an approach.

Most modular executives described in the literature, e.g. PACER[1], GEMCS[3], CHESS[4] and FLOWTRAN[5], employ the same fundamental information handling algorithm. They differ only in their degrees of sophistication, sizes of equipment subroutine libraries, etc. The systems are well suited for simulation of process performance as well as for equipment sizing and costing and have been used in plant improvement and optimization. However when they are examined from a synthesis viewpoint it is seen that they have virtually no creative capability. They are limited to user-supplied flowsheets and for the improvement of plant configuration or particularly for the evolution of a new plant configuration the approach is inefficient. Improvement must be gained by what is largely a trial and error process of successive evaluation of process configurations and this can in no way be guaranteed to arrive at the best attainable configuration. An example of design by this method is reported by Batstone and Prince(6) in planning steam systems for sugar refineries. Especially for design purposes, the development of a capability for process synthesis or automated flowsheet generation is desirable; in fact it is the next logical stage in the evolution of the modular systems approach to process design.

1.2.2 Synthesis

The sequence of decision making steps required for the complete synthesis of any chemical process has been detailed by Siirola and Rudd(7). They describe twelve steps alternating between synthesis and analysis, which lead from a given chemical reaction path through to the evolution of the final process flowsheet. Nine of these steps are implemented by their AIDES (Adaptive Initial DEsign Synthesizer) program which combines the computer capacity for systematic analysis with an intuitive capability provided through program interaction with the design engineer. AIDES is capable of proceeding through to the identification of the various processing tasks which together determine a basic processing scheme. The present study is mainly concerned with proceeding beyond this point to the implementation of the final synthesis steps, in particular to the "Task Integration" and "Final Evaluation" stages. These involve the actual selection and arrangement of the processing equipment to produce an optimal process flowsheet.

The selection and arrangement of equipment is essentially a discrete, combinatorial problem. It involves a choice between a very large but finite number of possible configurations which satisfy the specified processing objectives. It is necessary to select the configuration which meets some optimality criterion while at the same time being both feasible and operable.

Several recent approaches to optimal synthesis have dealt with the heat exchanger network problem which is briefly stated as follows. Given a number of hot and cold streams with given inlet conditions and outlet temperature specifications, construct the heat exchanger network which meets these requirements at minimum cost. To date studies have concerned only streams which have constant specific heats and transfer only sensible heat. Three such studies are described below.

Kessler and Parker[8] used a modified integer programming formulation (1 representing exchange between two streams, 0 representing no exchange) in which stream heat loads were divided into heat "elements" of finite size in order to linearize the network cost objective function. Satisfactory results were obtained for problems with up to 6 streams with a total of 28 elements. However the optimal solution must in general be dependent on element size and in practice the number of elements necessary to approach continuity in heat loads and thus remove this dependence may well make problems prohibitively large. Further, the strict mathematical formulation does not readily permit the flexibility of later approaches.

Masso and Rudd[9] introduced "HEURISTICS" to the problem. These heuristics are otherwise known as decision rules or rules of thumb. They are empirical rules embodying perhaps intuition or experience which are useful for problem decision making but are unproved or incapable of being proved. There may be some confusion regarding the association of the term heuristic with a learning process. There is in fact such a learning element in Masso's work but further use of the term heuristic in this study does not necessarily imply any such association.

In Masso's approach the network is constructed exchanger by exchanger, assigning new stream matches at each stage by using a set of heuristics. An example of such an heuristic is to select from those available that match which has minimum cost. Weighting functions were associated with each heuristic at each stage to build up experience on heuristic selection. This provides the program with a learning capability whereby it may move towards an optimal solution. The method has the advantages of simplicity and flexibility with the opportunity to incorporate useful empirical design rules within the heuristic set. However this dependence on the heuristic set used precludes any guarantee of optimality. The convergence rate of the iterative learning process is also dependent on the heuristics chosen and Masso has been unable to show that heuristic experience can be usefully transferred from one problem to another.

A more promising approach, since it does guarantee optimality, is the "branch and bound" method of Lee et al.[10]. It begins by generating all possible combinations of exchange to create a very large combinatorial problem.

The extraction of the optimal configuration then proceeds by branch and bound which is a very general technique from the field of operations research. It decomposes the original combinatorial set into (branches to) sets of much smaller and thus more easily solved sub (bounding) problems. With its guarantee of optimality, mathematical simplicity and generality the technique is a very attractive one. Branch and bound is in fact the optimizing technique to be used in this study and a more detailed description is given in section 2.1.

Another approach to synthesis is the "evolutionary" one developed by King et al[11] in their studies of separation processes. The evolutionary approach makes extensive use of heuristics. It starts with a basic user-supplied process flowsheet. This is then improved during an iterative sequence in which sets of heuristics are used both to isolate a process component to be improved and to suggest an appropriate improvement. The approach probably more closely follows the human designer's decision making process than do any of those above. It is a very practical one which allows the incorporation of a maximum amount of prior knowledge and experience but the usual heuristic-dependent limitations apply. The authors describe applications to an ethylene plant demethanizer column and a methane liquefaction process. The heuristic logic is automated only in the latter. The approach has been extended to the more general aspects of separation process synthesis in a very recent paper by Thompson and King[12].

1.2.3 Optimization

The sizes of large system optimization problems can still become overwhelming even if efficient solution algorithms are employed. In such cases it may still be possible to solve the problem by decomposition methods. These entail making use of the process structure to decompose it into a set of sub-processes which give rise to smaller, more readily soluble sub-problems.

Lasdon[13] has described such a method, for continuous process optimization problems, in which process decomposition is achieved by means of the assignment of transfer "prices" to flows between sub-processes. Prices are shown to be generalized Lagrange Multipliers. Sub-processes are then optimized with sub-process demands and productions free to float as additional decision variables, i.e., each sub-problem must decide on the quantities of inputs to be "bought" and outputs "sold" at the assigned transfer prices. Such provisions render the sub-process problems independent of the remainder of the process structure. The optimization algorithm then is a two level one with independent solution of the sub-problems at the lower level, while at the upper level prices are adjusted to reduce excess demands or supplies for flows connecting sub-processes. The overall optimum is reached when all such excesses have been reduced to zero. Overall convergence is not assured and may be slow particularly when there is strong physical sub-process iteraction. A further disadvantage is that the dimensional improvement (the reduction in number of problem decision variables) achieved through decomposition may be partially lost due to the additional decision variables introduced into each sub-problem. Process applications have been reported by Brosilow and Nunez[14] and Gembicki[15] and the latter has incorporated the algorithm into a modular process optimization system.

The present study is however concerned with discrete optimization problems. Everett[16] has shown that the generalized Lagrange Multiplier formulation for constrained optimization makes no restriction on the nature of the functions involved. Thus the approach is equally valid when the decision variable set and the objective and constraint functions are discrete. He shows that the method is especially useful for resource allocation problems where the resources can be committed to a number of independent areas (cells) and the overall payoff is merely the sum of the payoffs from each cell. This cell problem is in fact of the type considered by Lasdon. Everett describes an application to such a non-linear, integer allocation problem.

1.3 Synthesis - Study Philosophy and Objectives

This section is concerned with placing the present study within the broad general area of process synthesis and with defining the study objectives.

1.3.1 Study Philosophy

Of all the stages in process synthesis described by Siirola and Rudd[7] those of most concern to the practising design engineer are probably the final steps which result in the evolution of the process flowsheet. Frequently a basic processing concept will already be available to the engineer, whether it is from an existing process which is to be modified or from basic research or pilot plant studies for a new process. This concept may perhaps take the form of a reactor scheme or sequence of separation steps or both. The synthesis steps required to transform such a basic concept into a complete, operable process are in Siirola and Rudd's terminology, Task Identification, Task Integration and Final Evaluation. Tasks may take the form of requirements for stream temperature, pressure or phase changes, for component separations or for stream mixing or splitting. It is for the satisfaction of these requirements that equipment networks must be synthesized. The complexity of such networks will depend primarily on two factors. The first is the range of resources available to perform the tasks and the second is the degree to which it may be possible or required to integrate tasks by using some tasks to drive their inverses, e.g., a heating task driving a cooling task. Special regard must be given to problems of process feasibility, control and start-up, especially as the degree of process integration increases. The usual economic criteria for the worth of a design take no account of whether or not a process is practically operable.

The present study is concerned in particular with the complex problems of Task Integration. A more specific objective is to create a capability for automated process flowsheet generation within the framework of the modular approach which has been found so suitable for process evaluation or simulation. Such simulation systems are a very useful starting point for development and provide useful guidance as to data structures and equipment representation. Chemical processes represent a great diversity in processing concepts and equipment functions embodied in them. For this reason it is not considered practicable at this early stage in the development of process synthesis techniques to attempt to deal with completely general process concepts. This is especially true if a system is to be capable of the depth and detail necessary for the creation of very realistic process designs.

More specifically this study deals with synthesis of energy exchange systems which involve mainly stream temperature and pressure requirements. The important requirements involved in species transformation (reaction) and separation are more often embodied in the initial synthesis stages, i.e., the invention of the basic process concept which concerns the selection of the major equipment units. In this context energy exchange networks can be regarded as supporting equipment networks which satisfy processing needs external to the major equipment units. Nonetheless the efficiency of the supporting network in recovering process energy is often a vital factor in the overall process economics, and it may require a very high degree of process interaction and complexity of equipment interconnections. Energy exchange networks can thus provide a very useful area for development and application of synthesis techniques.

1.3.2 Study Objectives

The system to be developed in this study is to proceed through the following distinct three stages of synthesis.

1. Analysis of a basic processing scheme to identify a set of streams with unsatisfied temperature and pressure demands.
2. Generation of all possible equipment networks which satisfy these demands.
3. Extraction of the optimal network.

The system is to be built around the branch and bound combinatorial optimization technique. The reasons for its choice, as outlined in section 1.2.2, are mainly its guarantee of optimality, mathematical simplicity and generality and freedom from any true iteration scheme. It is most important in the broad area of optimal synthesis that the solution techniques themselves impose as few constraints as possible on the generality of solutions which may be obtained. Branch and bound is currently felt to be the most flexible in this regard. The incorporation of heuristic decision making into the branch and bound structure is very easy and extensive use will be made of it. This inclusion of heuristics may destroy the guarantee of optimality. However the heuristic capability to incorporate realistic and otherwise unusable design experience into the logical synthesis framework is a valuable, almost essential one in generation of realistic designs. In practical processes the concept of strict mathematical optimality is in any case difficult to apply as important additional factors such as controlability are much more difficult to rate than are the usual economic considerations.

A discrete, price-oriented, process decomposition algorithm is also to be included in the system. It is capable of achieving substantial reductions in sizes of discrete optimization problems by decomposition of the process into a number of smaller independent sub-process problems. The pricing structure imposed on flows between sub-processes is to be used to determine optimal interconnections between sub-processes. The algorithm can, conversely, be used for integration of a number of independent processing units which may function more efficiently as a single process. The decomposition technique can greatly extend the size of process which can be handled by the system.

It should be noted that in general this study is not concerned with equipment parameter setting. This falls into the area of continuous optimization, techniques for which are already well developed.

The above techniques are to be demonstrated by application to the design of ethylene recovery plants, commercially very important processes, in which efficient energy recovery is vital to the overall process economics.

CHAPTER 2

THEORY

There are two major areas of optimization theory which require further description and/or development for the present study. They are branch and bound and process decomposition, covered respectively in the following two sections.

2.1 Branch and Bound

This section reviews the work of Lee et al.[10] on the branch and bound optimization technique.

i) General

The branch and bound method is one of the most general approaches to the solution of constrained optimization problems. Its mathematical foundation can be simply expressed in terms of a bounding and an optimality condition, as follows. Start with an optimization (maximization) problem, A, which is excessively difficult to solve. The problem may be able to be replaced by branching to a problem or set of problems, B, which is related to but is much more easily solved than A. To be useful B must satisfy the following bounding condition. If the optimal solution to A were available and applied to B, that design must be feasible for B (i.e., must satisfy all technical constraints), but not necessarily optimal for B. Then if it also exhibits an equal or greater objective function value for B than for A, B is a valid (upper) bound for A. This bounding condition is expressed in (1).

$$O_B(D_A) \geqslant O_A(D_A) \qquad (1)$$

where O(D) is the objective function to be maximized for design problem D.

Note that (1) also implies that every feasible solution for A is also feasible for B.

Now if the optimal solution for problem B is found and is feasible for A and gives equal values of the objective function when applied to both A and B, then it is also the optimal solution to the original problem, A. This optimality condition is expressed in (2).

$$O_B(D_B) = O_A(D_B) \qquad (2)$$

Thus (1) and (2) guarantee that D_B is the optimal solution to problem A, and a very difficult problem has been solved through the solution of a much easier alternative problem.

The only difficulty in the application of the method is that of inventing appropriate bounding problems for particular situations - the basic strategy provides no guidance as to their selection, merely conditions which they must satisfy.

ii) Application to Heat Exchange Networks

Lee's strategy in applying branch and bound to the synthesis of optimal heat exchanger networks is described as follows.

Consider the problem of designing an optimal heat exchange network to satisfy temperature specifications for m given streams. These streams are conveniently classified as "hot" (to be cooled) or "cold" (to be heated).

They are to be series processed, contacting each with a sequence of other process and/or service streams until specifications are met.

Branch and bound takes a combinatorial approach to the problem. A useful bounding problem is first created by temporarily relaxing the network FEASIBILITY criterion. For a network to be feasible it is merely required that no stream be used more than once. Relaxing this criterion and thus allowing multiple stream use greatly simplifies the problem since it leaves the user free to formulate all possible STREAM MATCHES (matching of hot/cold stream pairs for heat exchange) without regard to the feasibility of any network created through any combination of these matches. Starting from the m primary or original streams, stream matching for exchange is begun. For each match the extent of exchange is fixed, i.e., it will proceed either until one stream is completely satisfied or until a certain minimum approach temperature is reached. Thus most matches produce residual (partially processed) streams which are then free to match with any other suitable streams. The matching process is continued until there are no further unsatisfied residuals. Stream matching information is used to build up sets of STREAM PROCESSING PATHS (sequences of matches/exchangers), one set for each primary stream. Costs are summed for all such complete processing paths, each of which represents one possible complete processing sequence for the primary stream in question. Note that it must be required that each path itself must be feasible (involve no multiple stream use); however any combination of paths, one per primary stream, which together form a possible network, may not necessarily be feasible.

For the m primary streams, if there are for each n_i $(i = 1,m)$ possible processing paths p_{ij} $(j = 1, n_i)$, then the number of possible networks that can be formed through combination of these paths is

$$N = \prod_{i=1}^{m} n_i \tag{3}$$

Although the majority of such networks may be infeasible, the task of merely testing the N networks for feasibility may be prohibitively large. This situation can be greatly improved by further implementing the branching strategy as follows.

Now branch to a set of bounding problems, each of which is defined to contain a certain specified stream match. One further problem, which excludes all such matches in the set, is added. This produces a set of problems which mutually bound or completely contain at least all feasible networks in the original set. By applying the network feasibility criterion to the problem at this stage (i.e., excluding all matches which require multiple use of any streams involved in specific bounding problem matches) a great number of infeasible networks can be immediately and efficiently eliminated. This efficiency is due to a "magnification" effect described as follows.

Each bounding problem must contain the specified stream match on which it is based. This allows elimination of any other matches which involve either of the streams in the specified match. The rejection of each such match may lead to elimination of a number of paths which contain the rejected match. The effect is further magnified since the rejection of each such path may lead to elimination of a still larger number of path combinations (networks) which contain the rejected path.

Thus the sizes of each of the bounding problems can be considerably reduced to the point where the bounding problem set is jointly a much smaller problem than the original, N. Further levels of branching can be made from each current bounding problem. This should proceed until the sizes of the problems are sufficiently small. Then they can be solved directly by sorting the network costs for each into increasing order and moving down this cost list until a feasible network is found. The minimum of all of these final level problem solutions is the overall optimum. The general branching strategy is shown in Figure 1 in which each node is associated with a specific stream match.

Example -

For example, consider the 4 stream problem (streams 1, 2 hot, 3, 4 cold) described by Lee et al.[10]. The stream matching process produces a total of 34 (10 + 5 + 7 + 12) primary stream processing paths which combine to produce a total of 4200 (10 x 5 x 7 x 12) possible networks. In the process 30 residual streams are created. The first level of branching problems is based solely on primary/primary matches except for the additional problem in which all streams are satisfied by services. The branching structure is shown in Figure 2, which also gives the sizes of the individual branching problems. A considerable reduction from 4200 is already evident and after a further level of branching the maximum individual problem size is reduced to 8, with a total of 55, at which stage problems are very readily solved by hand.

As an example of the way in which the magnification effect described above leads to this efficient reduction in problem size consider 1/4 sub-problem shown in Figure 2. The requirement that the 1/4 match must be included leads to immediate elimination of 8 matches which are incompatible with it. Elimination of all paths containing any one of these matches removes 24 paths, leaving a total of 10 paths (1 + 2 + 4 + 3) out of the original 34. These remaining paths combine to give a total problem size of only 24 (1 x 2 x 4 x 3) compared with the original 4200.

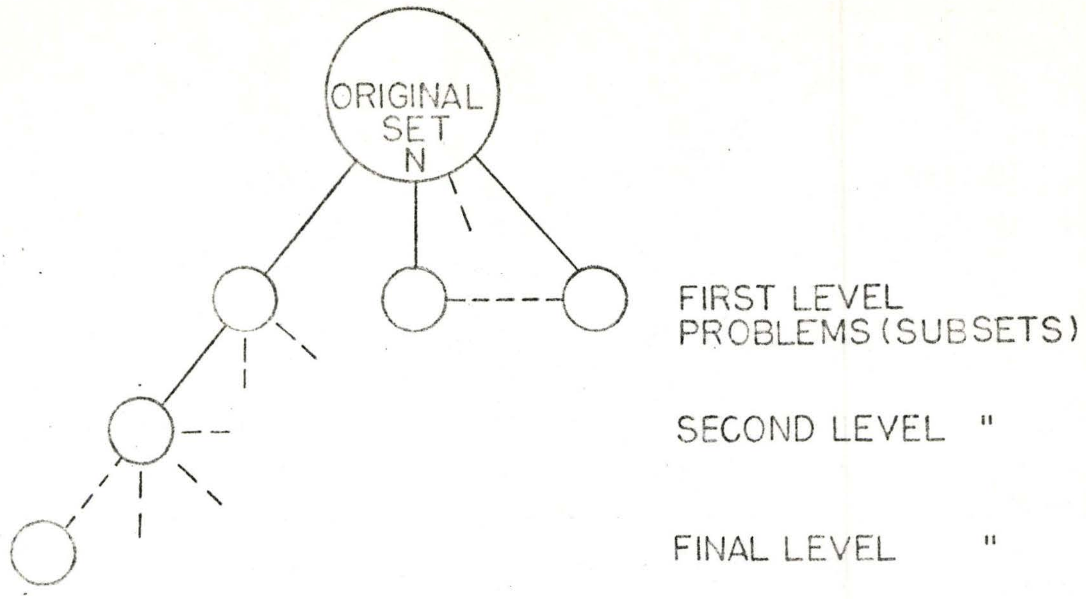


FIGURE 1. BRANCH AND BOUND-BRANCHING STRATEGY

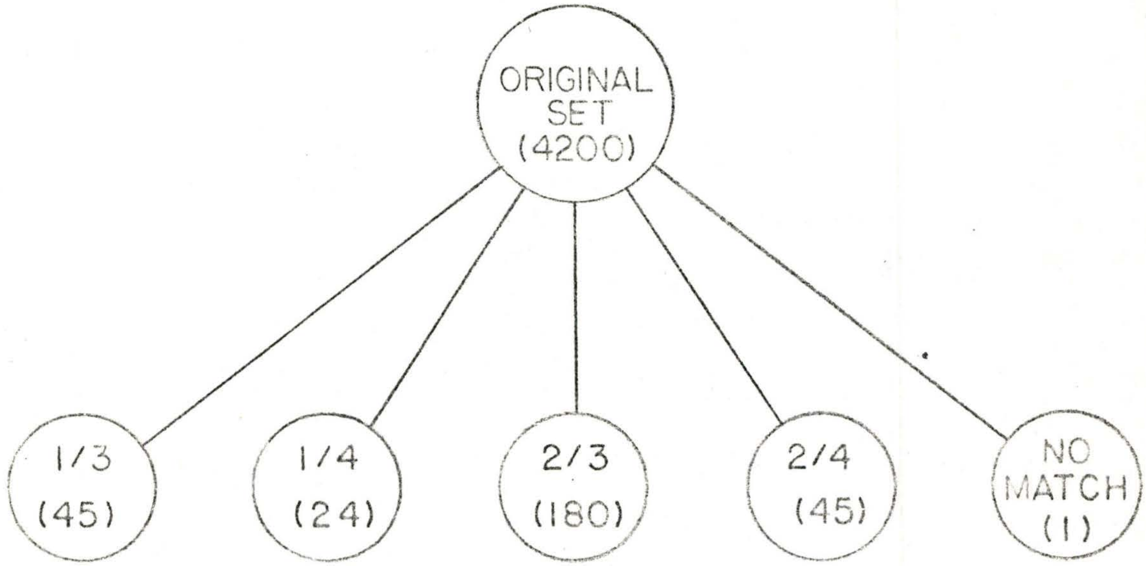


FIGURE 2. EXAMPLE FIRST LEVEL BRANCHING STRATEGY

Further refinements -

Further refinements to the basic strategy can be introduced by making use of the current best network cost to reject all paths which must lead to higher cost networks and by including a procedure to find a good initial feasible network. The reader is referred to Lee et al.[10] for further details. The efficiency of the branch and bound method is best reflected in the maximum size of sorting problems produced at the lowest level of branching. This is dependent on the choice of bounding problems as is seen in a later section.

2.2 Process Decomposition

It will be remembered from section 1.2.3 that decomposition is a technique which may be used to reduce the sizes of large system optimization problems by decomposing the problem concerned into a number of smaller more easily solved problems. This section shows how decomposition methods may be applied to the present type of discrete process design problem. For background the reader is referred to the work of Lasdon[13] and Everett[16].

Consider the process represented in Figure 3. The overall process is to be optimized by choice of a set of decision variables, M, associated with it. It has been divided or decomposed into two interconnected sub-processes each with its own subset of decision variables, $m \epsilon M$. The aim is to show how the overall process may be optimized by independent optimization of the sub-processes. This decomposition strategy, as will be seen later, leads to very substantial reductions in problem size as well as to more practical benefits in terms of limitations on process interaction. The present problem is concerned with the discrete choices involving equipment selection and arrangement so that the decision variable set, M, is both discrete and finite, i.e., there are only

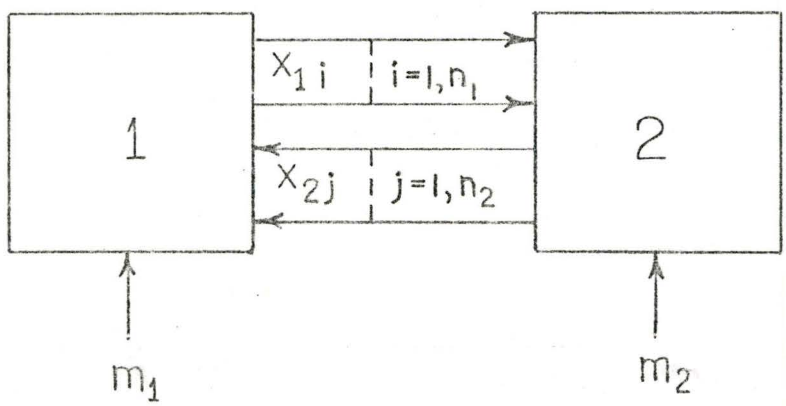


FIGURE 3. SUB-PROCESS INTERACTION

a finite number of discrete choices for the manner in which processing equipment may be assembled to fulfil the stream processing requirements. The interconnecting flows, $\underline{X}$, are dependent on M and thus are also discrete. They represent flows of intermediate streams transferred between ("sold" to or "bought" from) sub-processes. Other feed and product streams need not be shown.

The problem is of the "cell" or separable resource allocation type described by Everett[16] or the discrete analog of the continuous decomposition problem dealt with by Lasdon[13]. The resources concerned here are the internal flows, $\underline{X}$, which must thus satisfy equality constraints. Hence the problem is one of optimal discrete allocation of internal resources or, in terms of the sub-processes, the determination of optimal (stream) interconnections between them.

The original problem is

$$\underset{M}{\text{Minimize}}\ F(M) \tag{4}$$

or in terms of the sub-processes, since the problem objective function is separable,

$$\underset{m_1,m_2,\varepsilon M}{\text{Minimize}}\ F = [f_1(m_1) + f_2(m_2)] \tag{5}$$

where F is the overall process cost function and f are those for the sub-processes.

Consider the transfers between sub-processes

$$X_{1i} = X_{1i}\ [m_1, (X_{2j}, j=1, n_2)],\ i=1,n_1 \tag{6a}$$

and

$$X_{2j} = X_{2j}\ [m_2, (X_{1i}, i=1, n_1)],\ j=1,n_2 \tag{6b}$$

The sub-processes can be made independent by assigning to each X, a price, P, which is actually a generalized Lagrange Multiplier. Then the independent sub-process optimizations can be stated as their corresponding Lagrangians, (7),

$$\underset{m_1}{\text{Minimize}}\ f_1' = f_1(m_1) + \sum_{i=1}^{n_1} P_{1i} X_{1i} - \sum_{j=1}^{n_2} P_{2j} X_{2j} \tag{7a}$$

and

$$\underset{m_2}{\text{Minimize}}\ f_2' = f_2(m_2) + \sum_{j=1}^{n_2} P_{2j} X_{2j} - \sum_{i=1}^{n_1} P_{1i} X_{1i} \tag{7b}$$

noting that

$$f_1' + F_2' = F \tag{8}$$

provided that the flows, $\underline{X}$, are continuous across the sub-process boundaries.

For a given set of prices, $\underline{P}$, each sub-problem can be solved by branch and bound combinatorial optimization to yield optimal sub-process configurations. The mathematical advantage of decomposition is now obvious, since without it, the size of the combinatorial problem for the overall process is the <u>product</u> of those for the sub-processes. With the correct set of prices, $\hat{\underline{P}}$, the problem is decomposed such that the sum of the independently optimized sub-process solutions is guaranteed (the proof is given by Everett[16]) to give the overall optimum of the problem, or

$$\hat{f}_1' + \hat{f}_2' = F^* \tag{9}$$

where

$$\hat{f}_1' = \underset{m_1}{\text{Min}}\ f_1'\ (\hat{\underline{P}}),\ \hat{f}_2' = \underset{m_2}{\text{Min}}\ f_2'\ (\hat{\underline{P}}) \tag{10}$$

and F* is the overall process optimum, the solution to the original problem, (4).

The problem then is to adjust $\underline{P}$ in such a way as to move towards $\hat{\underline{P}}$. As Lasdon[13] has shown, for $\underline{X}$ continuous, $\underline{P}$ can be adjusted by deliberately creating a discontinuity in $\underline{X}$ between sub-processes and introducing $\underline{X}$ as additional decision variables for the sub-processes. Then $\underline{P}$ is adjusted to reduce excess supplies or demands for $\underline{X}$. This has certain disadvantages, as seen in section 1.2.3, of introducing convergence problems and sacrificing some of the reduction in dimensionality achieved through decomposition. In any case, in the present study, it is inconsistent with the discrete formulation of the problem to allow any such discontinuity in $\underline{X}$. Further it is seen that the solutions to (7), since they represent optimal sub-processes, are always in themselves feasible, i.e., they do not involve any multiple stream use or violate any constraints. Then with no discontinuity in $\underline{X}$ between sub-processes, there is the advantage that the overall solution $F = f_1' + f_2'$, is feasible if not necessarily optimal. This will always be the case where only two sub-processes are involved since it is obvious from Figure 3 that the equality constraints on $\underline{X}$ must always be met.

In the more general case, where a number of sub-processes are competing for the same resource, X, constraint violation, i.e., multiple use of X, is possible unless prices are correctly adjusted.

In order to proceed further, consider the dependence on the price vector, $\underline{P}$, of the overall process cost function, F. It can be seen that for this discrete system, F will be discontinuous with respect to choice of prices, $\underline{P}$; F in fact is piecewise constant in $\underline{P}$. This is because prices are artificial, internal variables and a change in price will not produce a change in overall process cost unless it produces a change in network configuration with corresponding change in flows, $\underline{X}$. Thus there will in general be a certain range of $\underline{P}$ around $\hat{\underline{P}}$ within which the overall optimal solution F* will be constant. It is necessary only to be within this range to solve the overall problem and this permits a certain amount of flexibility in price adjustment.

Everett[16] has suggested in the solution of such discrete cell problems, that prices be adjusted by trial and error or by searching over a pre-determined grid. In this way solutions can be produced over a range of $\underline{P}$ and the optimum extracted from them. This is the method to be used in this study. However as will be seen later, it is possible to obtain close estimates of prices from a physical standpoint. Through this technique the problem of dimensionality in the choice of $\underline{P}$ can be substantially reduced by using a common scale for pricing streams of a similar nature.

In general unless an infinitely small grid is used the best solution obtained cannot actually be guaranteed to be the global optimum but good feasible solutions can always be generated. In fact generation of a range of process configurations may be an advantage, especially if there is little cost difference between them. Then other more practical criteria, related to process operability may be applied to select the "best" process configuration.

More specific details of the costing scheme for the particular process examples considered in this study are given in section 3.3.

CHAPTER 3

DESIGN CONSIDERATIONS

In order to be able to synthesize realistic process networks it is necessary to supplement theory with more practical process-oriented considerations. This study is concerned with the synthesis of energy exchange networks, in particular as applied to low temperature gas separation processes where the efficient recovery of low temperature thermal energy is particularly important. Thus many of the design considerations to be developed in this section will tend to be specific to this type of process. These considerations may be described in three sections, the selection of equipment or unit operations, the development of design rules or heuristics and stream energy pricing considerations.

3.1 Unit Operations

Any energy exchange network is to be synthesized from a basic set of unit operations or process equipment. Those for the present study are listed below.

i) Countercurrent heat exchange

ii) Polytropic single stage compression

iii) Adiabatic (valve) expansion

iv) Adiabatic stream mixing/splitting

Further, there are certain instances where it is desirable to provide a standard pre-coded assembly of unit operations, termed "sub-process procedures".

Thus the present synthesis system provides for multistage compression with water intercooling and for vapor recompression condensation/reboiling. A special case, described in detail later, is the refrigeration routine which is coded as a skeleton flowsheet generator with some limited decision making capability. With the exception of the refrigeration unit all unit operations and sub-process procedures are shown symbolically in Figure 4.

3.2 Heuristic Development

For most processes it is possible to draw up a list of relevant design considerations. These may vary widely in form. They may range from the very general to rather specific, from being highly empirical to being theoretically justifiable. However they may be broadly categorized as relating to -

i) Processing objectives

ii) Operating objectives - control during start-up, shutdown or steady operation

iii) Thermodynamics

iv) General design experience

Where possible these considerations can be translated into a set of logical, programmable design rules or "heuristics". These may then be used to considerable advantage in setting the order and extent of unit operations and particularly in pre-screening of prospective stream matches for heat exchange. Their use can greatly reduce unnecessary design effort and problem size and complexity. By their very nature these heuristics tend to be rather specific to certain processes or types of processes where similar objectives apply. However this capability of being able to incorporate design rules into the logical synthesis structure adds considerably to the flexibility and usefulness of the approach.

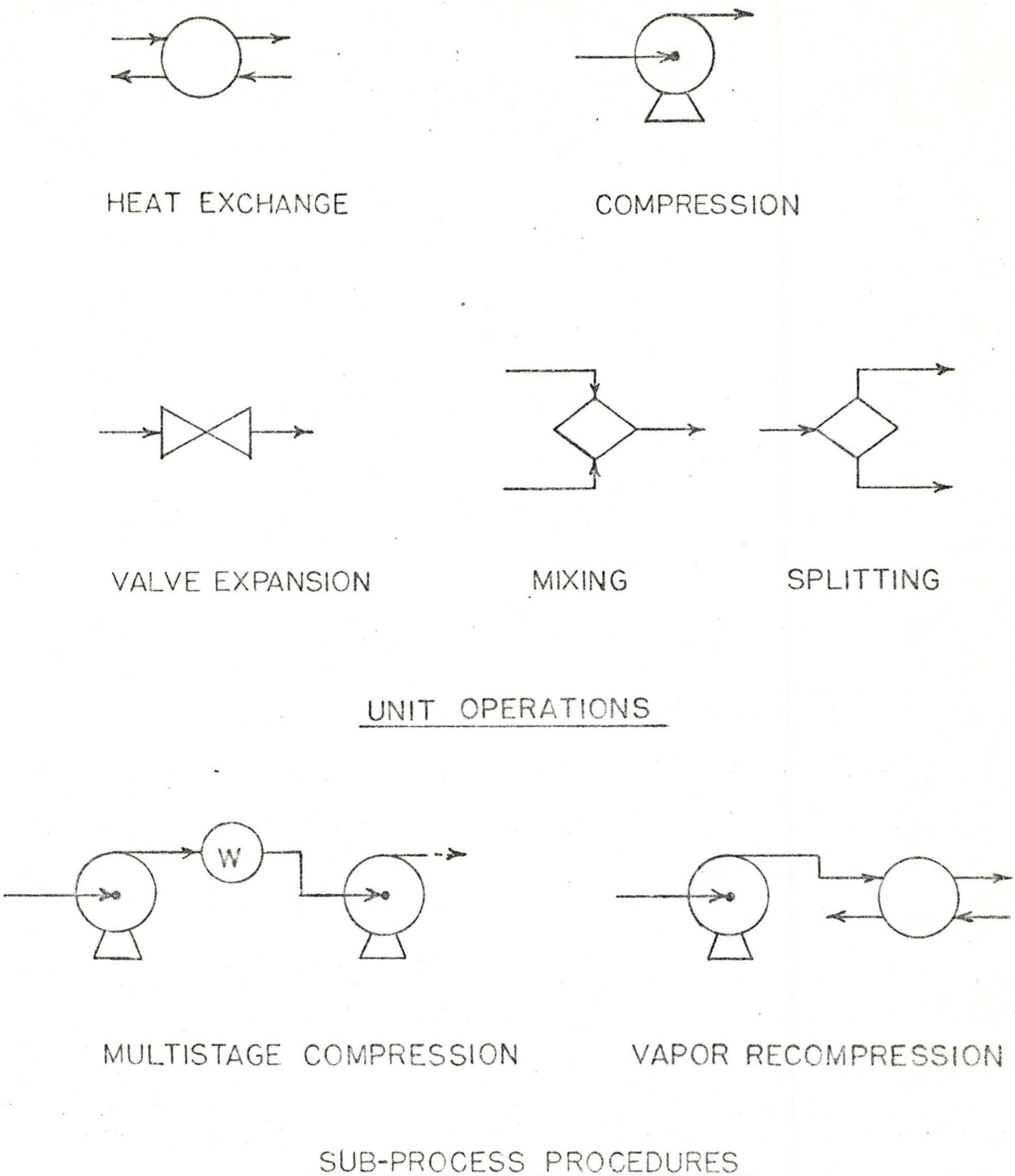


FIGURE 4. UNIT OPERATIONS AND SUB-PROCESS PROCEDURES

The major heuristics used in the present study are described below. Additional heuristics are introduced as required for application to specific processes.

Ordering Unit Operations -

i) Carry out all pressure change operations before heat exchange.

This is generally the rule for the type of low temperature gas separation process considered, where gases must be compressed in order to liquefy them and refrigeration must be recovered at the lowest possible temperature. This heuristic can be justified rather more generally as follows. The processing objectives for the present process type are largely concerned with thermal rather than pressure energy recovery. Thus pressure changing can be regarded as raising or lowering the thermal energy level of a stream in order to make technically feasible or to improve the thermodynamic efficiency of the subsequent heat exchange. Pressure change thus precedes heat exchange.

Extent of Unit Operations -

i) Set a minimum temperature of approach for heat exchange.

This is a practical limitation imposed by process equipment.

ii) For vapor recompression, compress just sufficiently to meet the above minimum approach in the subsequent exchanger.

iii) Limit the pressure ratio for a single compression stage.

This again is a practical equipment limitation.

Stream Matching -

i) Set a maximum entropy increase/BTU for process/process exchange.

This is aimed at minimizing heat exchange irreversibilities and thus conserving refrigeration and reducing overall energy costs. It is a particularly important consideration expecially in low temperature situations.

ii) Exclude vapor/vapor matches

This is necessitated by the low heat transfer coefficients in vapor/vapor exchange. These lead to high costs for recovering what are, since only vapor phase sensible heat is involved, usually only small quantities of energy.

3.3 Stream Energy Pricing for Process Decomposition

The stream pricing scheme employed for process decomposition/integration in the present study is developed as follows. Thermal energy recovery, particularly at low temperatures, is regarded as the prime consideration. Thus it follows that basic stream values or prices can be estimated as a function of temperature, i.e.,

$$\text{Price/BTU} = pr(T) \tag{11}$$

In the present case the form of the function is readily established from the real physical costs associated with service streams. Sold streams are classified as hot or cold with respect to cooling water, which serves as a convenient basis point for both temperature and cost. Remaining points are provided by steam on the hot side and actual refrigeration production costs on the cold side. To provide a continuous function for purposes of interpolation and integration, cubic splines[17] are fitted to both hot and cold sections. This can be seen in Figure 14.

An approach to stream pricing that is more theoretically based should be considered at this stage. This comes from the work of Tribus and Evans[18] on heat recovery in sea water desalination processes. They suggest the use of "exergy" or availability rather than energy as a stream pricing parameter, since it is exergy rather than energy which is consumed by process irreversibilities.

The exergy function is given by

$$\varepsilon = \Delta H - T_o \Delta S \tag{12}$$

Then for an incremental energy transfer at constant pressure, i.e., heat transfer, the exergy function, expressed on a unit energy basis, can easily be shown to be

$$\varepsilon/\mathrm{BTU} = \frac{T_o - T}{T} \quad , \text{ the Carnot fraction} \tag{13}$$

In the present case the sink temperature, T_o, is conveniently taken as that of cooling water. Thus the price function, (11), should be of Carnot fraction form for both hot and cold streams. The simplicity of the relationship is clearly attractive and its validity will be examined in light of computational results.

The stream pricing technique is a means of determining optimal stream interconnections between sub-processes, i.e., it determines whether a given stream is to be used within a given sub-process or sold to another. For this reason it is necessary to modify the price function to account for two further factors which affect the true value of a stream to any sub-process.

The first is the degree of irreversibility involved in stream usage. This depends on the temperature difference between the two streams during exchange; the higher the irreversibility the less desirable the match and the lower the true value of the stream in question. This factor can be accounted for by introducing a temperature displacement, δ, termed a "discount" parameter. It can be thought of as being representative of the actual temperature difference between the two contacting streams. Thus the price function, (11), becomes $pr(T \pm \delta)$. To reflect irreversibility the sign should be positive for cold streams and negative for hot streams, thus always in the direction of reducing value due to irreversibility. While δ itself should be positive to reflect irreversibility, it is in fact a parameter representing <u>relative</u> irreversibility between internal and external usage and thus can have either positive or negative value.

There is also a variation in the (equipment) cost of stream usage and though it is not strictly related to exchanger irreversibility, its effect is conveniently included in the discount parameter.

Thus the value of a stream between any specified temperature limits is obtained by the integration

$$P = \int_{T_1}^{T_2} pr(\theta \mp \delta)\left(\frac{dH}{d\theta}\right) d\theta \tag{14}$$

where $dH/d\theta$ is, for a single phase stream, just the specific heat.

The integration method is described in detail in Appendix I.1.

The δ parameter is thus used to adjust stream transfer prices, the basic prices being fixed by the form of the energy value spline(s). As seen earlier, prices are Lagrange Multipliers and strict optimality can only be guaranteed if all multipliers are adjusted independently, i.e., if there is one δ associated with each stream transfer. However if there are a large number of transfers, then the problem of dimensionality in the adjustment of the price vector, $\underline{P}$, may become serious. In this case it is suggested that a single δ be applied to a set of similar (hot or cold) stream transfers between any two sub-processes or even for all similar inter-process transfers. The number of adjustable parameters is then reduced from the number of transferred streams to the number of independent δs. This introduces the possibility of missing some solutions. However as the optimum solution has been shown to be constant over a certain range in $\underline{P}$ and the expected variation in δs is comparatively small, the risk is considered to be justified in terms of the reduction in dimensionality.

CHAPTER 4

PROGRAM SYSTEM

4.1 General

The techniques described or developed in earlier sections have been implemented in the form of a program system called OPENS (Optimal Process Equipment Network Synthesizer). In its present form it is oriented towards the synthesis of energy exchange networks required to satisfy process stream temperature and pressure demands. The particular process applications demonstrated are in the area of low temperature gas separation. However the concepts should be generally applicable to any similar energy exchange situation which can be formulated as a discrete, sequential processing problem. The synthesis steps accomplished by the system have been given earlier but will be repeated here in order to facilitate the description of the individual program functions within the system. They are:

i) Analysis of a given basic processing scheme to identify a set of streams with unsatisfied temperature and pressure demands.

ii) Generation of all possible equipment networks which satisfy these demands.

iii) Extraction of the optimal network.

The structure of the program system with its major elements is shown in Figure 5. It is, as will be seen later, a modularly oriented system, i.e., any network is synthesized from a combination of basic processing modules, represented within the system by equipment subroutines. The system borrows

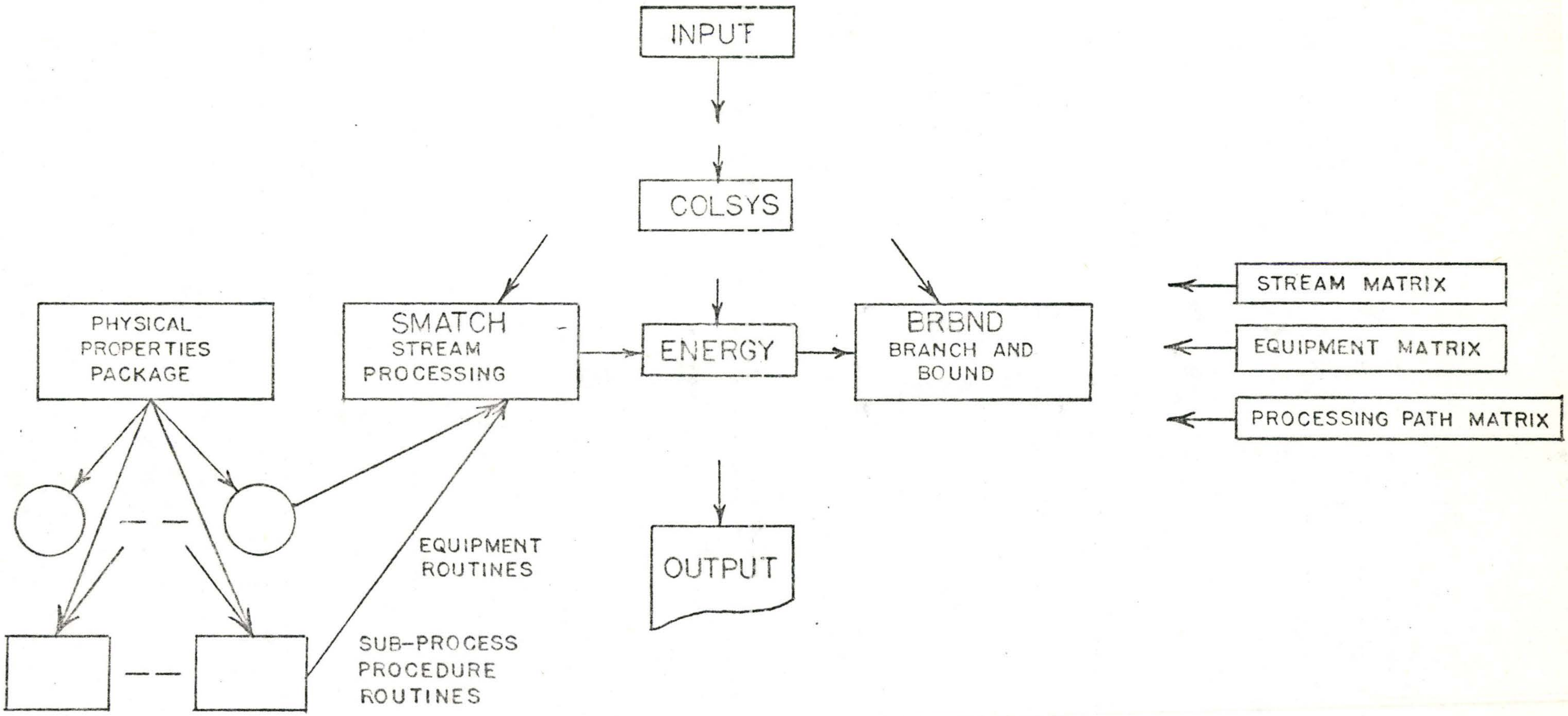


FIGURE 5. PROGRAM SYSTEM STRUCTURE

greatly from its simulation system predecessors, particularly with regard to data structures and equipment representation. However the realistic synthesis procedure demands decisions to be made that are specific to certain equipments in the available unit operations set. Hence this synthesis system is of necessity much more specific to particular processes than are comparable simulation systems. The major features of the system are described in the following section. Full program listings, graphical algorithms and sample data sets are given in Appendices II and III.

4.2 Program Functions

Descriptions of major program sections are given as follows.

4.2.1. Task Identification (COLSYS)

The first synthesis stage is the identification of process tasks, represented in the present case by unsatisfied stream temperature and pressure demands. It is carried out by COLSYS. Since the example processes studied are gas separation plants built around sequences of distillation columns, COLSYS is set up specifically to analyse such systems. It is essentially a small modular simulation executive which computes a specified column sequence, performing overall heat and mass balances and thus computing stream flows and conditions. Process tasks are identified by comparing supplied stream specifications with actual conditions. A special case is that of the column liquid and vapor reflux generation tasks which are created automatically within the program. Streams are classified as "hot" or "cold" (to be cooled or heated), a necessity for later stream matching.

The approach used here is rather specific to a certain class of processes and in general it may be necessary for the user to provide particular task identification routines. There is no restriction on the manner in which tasks are identified and it is possible that this stage may be accomplished outside the program system.

4.2.2 Stream Processing Path Generation (SMATCH)

Stream processing is handled by SMATCH which computes all possible equipment sequences to satisfy the stream temperature and pressure specifications generated above. As seen in section 3.2, pressure specifications are to be met first. These are satisfied by either (multistage) compression or expansion and since no alternatives are involved this "pre-processing" phase does not enter into the subsequent branch and bound optimization. The program then proceeds to satisfy all temperature specifications by exchange with other process streams or services (steam, cooling water or suitable levels of refrigeration). Only discrete, series processing is in general permitted, as limited by the formulation of the branch and bound technique. Exchange matching is continued until all specifications have been met for all primary (original) streams and their (partially processed) residuals. Vapor recompression is permitted between primary streams for which phase changes are indicated. In this case the compression and subsequent exchange steps are treated as a single stream match for optimization purposes. The sets of heuristics described earlier are used to determine the extent of equipment operations, and in particular to pre-screen each technologically feasible match in order to reject unfavourable matches a priori. Due to the wide variation in form that they may assume they are programmed into the routine rather than supplied in some fashion as input data. Stream matching information is built up as sequences of equipment numbers in the stream processing path matrix. A routine is included to ensure that each individual processing path remains feasible, i.e., uses no stream more than once. A stream sale is represented as a processing equipment in order to be compatible with the processing path data structure.

4.2.3 Stream Energy Costing (ENERGY)

Sold stream values are computed by ENERGY. The routine also selects appropriate refrigeration levels and computes costs for exchangers using refrigeration. Values for both process streams and refrigeration are obtained from the current energy value splines, as shown in Figure 14. Total costs for each processing path are computed after this energy costing step.

4.2.4 Selection of Optimal Network Configuration (BRBND)

The set of stream processing paths (equipment sequences) from SMATCH forms the primary input to BRBND, the branch and bound optimizing routine. Its task is to select the lowest cost feasible set of processing paths (one per primary stream) which jointly define the optimal process network configuration. It is essentially a computerization of Lee's branch and bound technique, as described in section 2.1. The program allows up to three levels of branching and automatically selects appropriate bounding problems. A routine is included which establishes a good initial feasible network in order to increase computational efficiency.

4.2.5 Physical Properties Calculation

Accurate equilibrium, enthalpy and compressibility values are supplied to the system by a modified version of the CHESS[4] simulation system physical properties calculation package. The package calculates mixture values from sets of 15 basic physical constants for each pure component. Equilibrium data are computed by the method of Chao and Seader[19] as modified by Grayson and Streed[20]. Vapor phase fugacities are obtained from the Redlich-Kwong equation of state. Enthalpies for both phases are based on zero pressure heat capacities as derived from the Redlich-Kwong equation, with liquid phase compressibilities supplied by the generalized equations of Yen and Woods[21]. The package supplies values for single phase streams only. Properties for the two-phase region are computed through a rigorous adiabatic/isothermal flash routine. This program, which also serves as an equipment routine, is also modified from the CHESS system.

4.2.6 Equipment Routines

Conventional simulation-type routines are used to size and thus cost all equipments. They are briefly described below.

The column model is based on the approximate pseudo-binary design procedure of Hengstebeck[22]. It makes the McCabe-Thiele assumption of constant molal overflow and uses constant relative volatilities to represent phase equilibria. It is much faster than conventional plate-to-plate methods and is capable of good accuracy as long as the constant molal overflow assumption is reasonably valid.

The exchanger routine uses a set of supplied film heat transfer coefficients corresponding to the phases of the contacting fluids. Overall coefficients are computed by addition of film resistances. The exchanger area is then computed by numerical integration with the total heat load divided into 10 equal increments.

The compressor model is based on a single stage polytropic compression process. The power requirements are estimated from the enthalpy at the computed outlet temperature assuming adiabatic operation.

The CHESS based rigorous adiabatic flash routine described previously is used as the adiabatic expansion routine. The same routine also serves as an adiabatic mixer.

Sub-process procedures are, as described in section 3.1, standard assemblies of unit operations. Such procedures for multistage compression and vapor recompression are included within SMATCH. The only independent sub-process procedure is that for the refrigeration unit. It is a small executive which generates the equipment sequence for a conventional cascade refrigeration unit[23] and is described in more detail in section 5.4.

Equipment costs are computed from standard "power law" relations with Lang factors to relate installed to delivered costs. Values were obtained from Bauman[24], Peters and Timmerhaus[25] and Hand[26]. A constant fraction of the total capital cost is amortized each year and added to the operating cost to obtain the total yearly process cost which is the objective function for optimization. Data for equipment and service costs as well as for other relevant system parameters are given in Table 1. Data for distillation columns are given separately in Table 2.

4.3 System Data Structures

The successful solution of large system problems of the type considered in this study depends largely on the use of efficient data structures. There is a large quantity of stream and equipment information which must be stored in very compact fashion yet must require a minimum of regeneration of necessary information. The major data structures for the present system are described below. It can be seen that they are loosely based on the comparable structures for modular simulation systems, but the nature of the synthesis procedure requires a certain amount of additional information. More specific details of system data structures are given in Appendix II.2.

i) Stream information is stored in a simulation-type stream matrix with separate sections for hot and cold streams. Two vectors are used for

Table I

General System Parameters

Equipment Costs [Installed Capital Cost = a*(size^b * Lang factor (f)]

Equipment	a	Size	b	f
Heat Exchanger	82	$\leq$ 400 ft^2	0.6	4.0
Heat Exchanger	25	> 400 ft^2	0.8	4.0
Compressor	480	HP	0.76	2.5
Compressor Motor	34	HP	1.0	2.5

Material Cost Factors for Heat Exchangers

Down to -50°F	Carbon steel	1.0
-50°F - 150°F	Nickel steel	2.0
Below - 150°F	Stainless steel	3.5
Amortization fraction	0.3/year	

Service Costs

Steam	\$1.00/1000 LB @ 365°F (150 psia)
Cooling Water	\$0.02/1000 GAL(IMP) @ 75°F - Temperature Rise 10°F
Electric Power	\$0.007/KWH

Other Parameters

Minimum Exchanger Approach	10°F
Maximum Compressor Pressure Ratio per Stage	4.0

Table 2

Distillation Column Parameters

Equipment Costs [Installed Capital Cost = a*(size)^b * Lang factor (f)]

	a	Size	b	f
Column Shell	14.5	Wt (lbs)	0.7	4.0
Trays	48.0	Diam. (ft)	1.7	4.0

Material Cost Factors and Stresses

		Cost Factor	Stress (psi)
Down to -50°F	Carbon Steel	1.0	13750
-50°F to -150°F	Nickel Steel	2.0	16000
Below -150°F	Stainless Steel	3.5	18750

Tray Efficiency	70% throughout
Tray Spacing	24" (18" for C_2, C_3 Splitters)
Corrosion Allowance (Carbon Steel)	1/16"

each stream; the stream control vector contains stream status and specification information and the stream properties vector contains normal properties and flow information.

It should be noted that both stream bubble and dew point temperatures have been added to the usual parameters as these values are frequently used in phase calculations. For the present series processing situation, where stream compositions are constant, these temperatures change only infrequently when stream pressures are altered. Thus significant computation time (around 0.1 seconds per bubble or dew point estimation) can be saved by carrying these values in the stream vectors.

The constancy of stream compositions permits another economy in storage, since compositions for a primary stream and all of its residuals can be represented by a single vector of stream mole fractions.

Within equipment routines stream property information is accessed through working vectors. Information transfer between the stream matrix and working vectors is handled by a stream moving utility routine.

ii) Each equipment is represented by a two section vector in the equipment matrix, containing (a) equipment number and type and inlet/outlet stream numbers and (b) size and cost information. An equipment working vector is used to transfer values to and from equipment routines.

iii) Stream processing paths are stored as sequences of equipment numbers in columns of the stream path matrix, which also contains total path costs. Each column contains a unique, complete processing path for that stream and is a very compact means of path representation.

As each new match is added to a given path a check must be made to ensure that no multiple stream use is introduced. This is made through stream "histories" each of which is a list of streams used in the evolution of the matched stream in question. These histories are generated from information in the equipment, stream and stream path matrices each time they are needed and thus a certain amount of data regeneration is necessary to achieve this compaction in storage. This approach should be compared with the original method of Lee et al.[10], which, although not computerized, did not use equipment numbers and maintained stream history information for all residuals. The present approach is felt to be less cumbersome and more easily understood by the user as well as requiring less total storage.

4.4 Programming and Operating Details

The OPENS system has been programmed in FORTRAN IV for the CDC 6400 computer. Both to allow user operating flexibility and to reduce storage requirements, the system has been run in three major batch sections, represented by COLSYS, SMATCH and BRBND. The refrigeration unit (RUNIT) which is described later forms a fourth section. With a data structure capacity for 100 equipments, 100 total streams (including residuals) and 200 processing paths, the maximum core storage requirement has been $50K_8$. The maximum computation time for any section for the process cases run has been less than 20 seconds.

PART II

APPLICATION STUDIES

CHAPTER 5

ETHYLENE PLANT - PROCESS DESCRIPTION AND CONSIDERATIONS

The two process applications to be presented in Chapters 6 and 7 are both to ethylene plant designs. The ethylene process is of growing importance to the petrochemical industry as the demand for ethylene as a basic chemical is now second only to that for synthetic ammonia. Ethylene production is an area for considerable interest and technological improvement and is thus the subject of a wealth of literature. These are not the only reasons for its selection. As will be seen later, the process has very high energy costs which make it a particularly suitable area for application of the synthesis techniques developed in this study.

5.1 Ethylene Plant Description

For purposes of process analysis, an ethylene plant may be divided into three main sections - cracking, purification and product recovery. A schematic of a typical process is shown in Figure 6. For descriptions of two modern ethylene plants the reader is referred to Clancy and Townsend[27] and Aalund[28].

5.1.1 Cracking

Ethylene may be obtained from cracking almost the whole range of petroleum fractions from ethane to crude oil. The choice of feed stock is a matter of economics depending on availability and, to a smaller extent, the market for the by-products. Ethane, propane, natural gasoline and naptha are

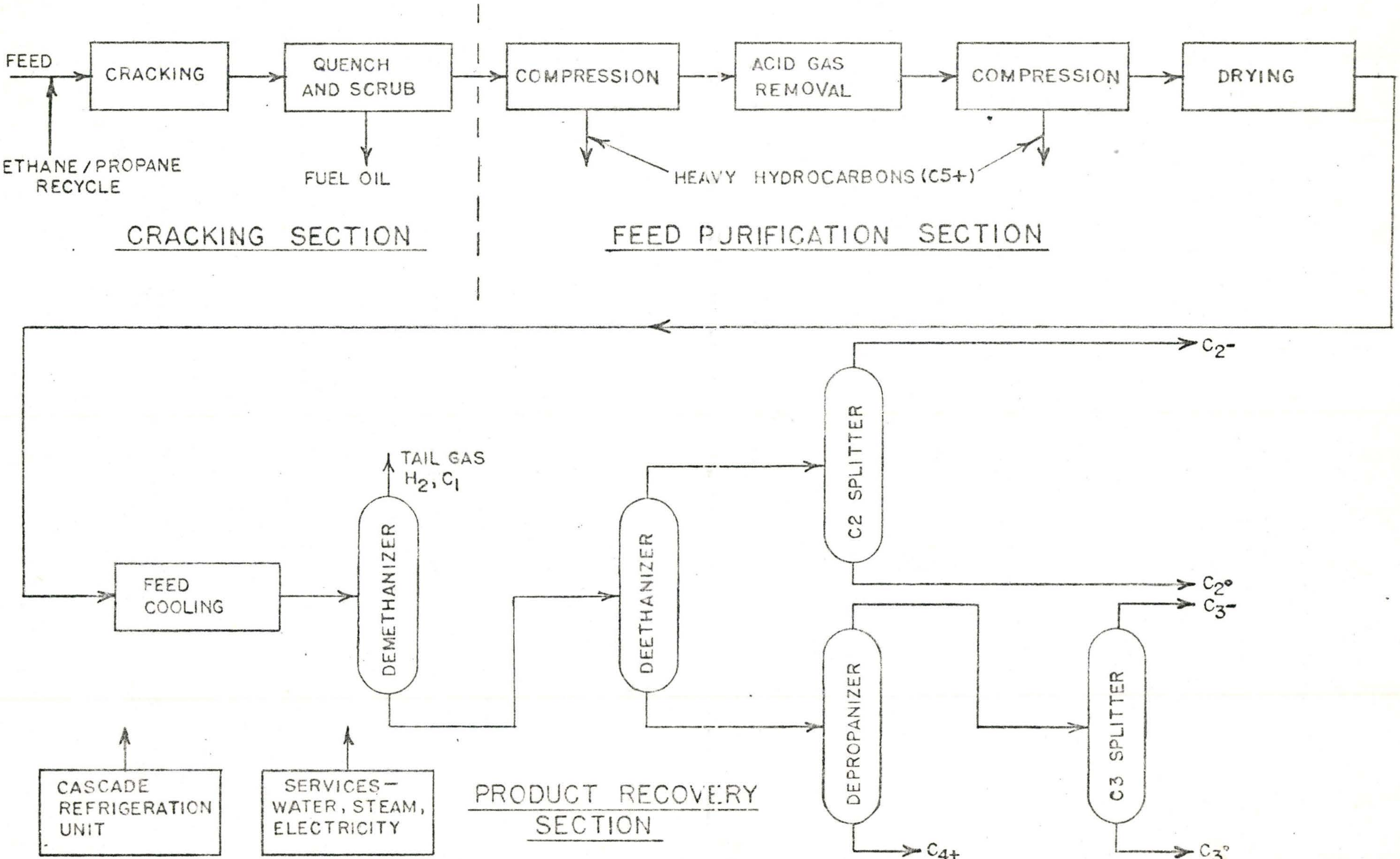


FIGURE 6. ETHYLENE PLANT TYPICAL SCHEMATIC

the most common.

The feed is first vaporized and mixed with steam before entering the cracking reactor which operates at high temperature and atmospheric pressure with short contact times. Steam addition serves several purposes. Firstly it lowers the hydrocarbon partial pressure thus favouring the equilibrium of the desired reaction; secondly it reduces reactor contact time lessening production of undesired products; finally it acts as a scavenger for some of the coke formed.

The reaction is arrested immediately by water quenching followed by scrubbing with either water or oil. The waste heat recovered by these two units is used to generate process steam at appropriate levels. Together with steam generated from cracking furnace flue gases, the total may be sufficient to supply all subsequent process energy requirements.

5.1.2 Feed Purification

In this section of the process impurities such as water and acid gases are removed prior to separation of the major hydrocarbon components. The cracked gas mixture is compressed in a multistage compressor train provided with intercoolers and separator drums. Water and some heavy hydrocarbons are partially removed. The gas is then scrubbed with caustic primarily to remove carbon dioxide and hydrogen sulphide. This stage may also be accomplished after some intermediate compression stage. Finally the last traces of water are removed by drying over alumina and/or molecular sieves. This is essential to prevent the formation of solid hydrocarbon hydrates in the low temperature recovery section.

5.1.3 Product Recovery

The product recovery stage is perhaps the most important and expensive and subject to the greatest degree of variability. It is also the process section with which this study is primarily concerned and will thus be described in some detail.

The gas stream from the purification section contains hydrogen and hydrocarbons from methane down to C_4s and heavier, the composition varying with the feedstock. The principal products required are (high purity) ethylene and propylene. Ethane and propane product streams are generally recycled to cracking reactors. There are two further products, a tail gas containing hydrogen and methane and a stream containing C_4s and heavier. These separations are achieved by conventional bubble-cap or valve-tray distillation columns. A minimum of five columns are required to obtain all of the above product streams. Separation conditions throughout the process may range as high as 565 psia for pressure and as low as -250°F for temperature.

i) Separation Sequencing

The separation sequence which has been shown in Figure 6 is only one of a number that may be employed. The ethylene-ethane (C_2 splitting) and propylene-propane (C_3 splitting) separations are the most difficult because of close component relative volatilities. Therefore they are always at the end of the separation scheme where the flows are smallest. The ordering of the other three separations is by no means standard and depends largely on cracked gas composition. The gas composition and separation order together determine the quantities and levels of refrigeration required in the process. As temperatures may be very low the refrigeration costs are frequently the determining factor in choice of separation sequence. These considerations are discussed by King[23] and Charlesworth[29].

Three alternative separation sequences are shown in Figure 7. Probably the most common is (a) as was shown in Figure 6, which is typical of plants cracking ethane and propane. Here demethanization is the first stage. A variant is shown in (b) where deethanization precede demethanization. If the demethanizer is placed after the deethanizer, (b), its feed is reduced to a minimum and the column becomes of minimum size. However all tail gas must then pass through the deethanizer increasing its refrigeration requirements. If the demethanizer is placed first, (a), it must be larger and will require additional refrigeration. However the deethanizer refrigeration requirements are greatly reduced with the elimination of the tail gas which is the reason that this sequence is normally preferred especially if the tail gas flow is large.

When the feedstock is naptha or natural gasoline configuration (c) is usually preferred. Here there are substantial quantities of C_3 and C_4 hydrocarbons in the cracked gas and placement of the depropanizer first enables these components to be separated before other steps which require low temperature refrigeration.

ii) Operating Conditions

The other major design decision is that of choosing operating conditions or more specifically operating pressures. In this regard ethylene plants can generally be divided into two categories, high and low pressure processes.

The high pressure process, most common in North America, involves demethanization at around 550 psia with subsequent separation pressures ranging down to around 200 psia in the C_2 splitter and 100 psia in the C_3 splitter. A typical plant is described by Aalund[28].

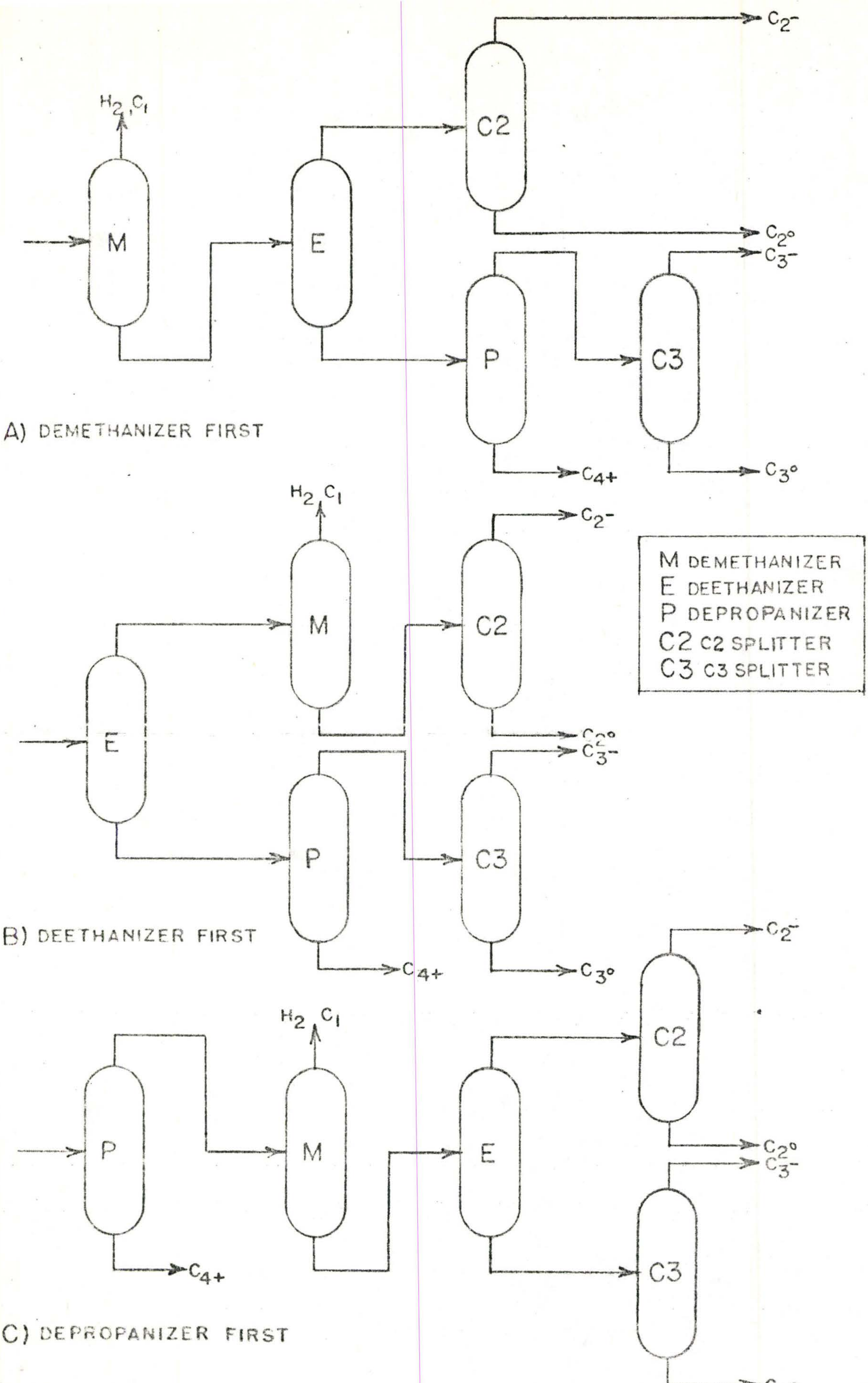


FIGURE 7. PRODUCT SEPARATION SCHEMES

The lowest temperature required is that to produce demethanizer overhead reflux. It must be low enough (around -160°F) to minimize overhead ethylene loss in the tail gas. At high pressures this can generally be achieved by ethylene refrigerant from a propane-ethylene or propylene-ethylene cascade system, perhaps supplemented by Joule-Thompson cooling with expanded tail gas[23].

The low pressure process, as described by Baldus and Linde[30] or Brooks[31], is more frequently used in Europe and is descended from liquid air technology. Separation pressures do not exceed 250 psia and range as low as 20 psia for C_2 splitting. This produces much lower temperatures (down to around -250°F for the demethanizer overhead) and requires the addition of a methane cycle to the refrigeration cascade. Inevitably refrigeration costs are increased but advantages result from lower feed compression requirements and easier separations due to increased relative volatilities at lower pressures. Baldus and Linde claim significant improvement in power consumption over the high pressure process. Features of both high and low pressure operation have been compared by Ruhemann and Charlesworth[32].

As will have been noted above, a most important requirement for product recovery is the provision of a large quantity of refrigeration. The refrigeration unit associated with the process is commonly a two or three section cascade compression system employing as refrigerants propane or propylene, ethylene, and methane if required. A number of different levels may be required from each section or circuit to satisfy process cooling and condensation requirements. A typical system is described by King[23] and a more detailed description will be given in section 5.4.

5.2 Process Energy Considerations

Since it is the product recovery section which is to be the object of the application studies, the energy recovery aspects of this section of the process should now be considered in some detail.

A large component of ethylene production cost is associated with power consumption, mainly for feed and refrigerant compression. Assuming electric compressor drivers, the total power requirement is around 1350 kWh per ton of ethylene and compressors make up the largest item of capital expenditure. Thus the efficient utilization and recovery of energy is of prime importance and much recent technological effort has been expended in this direction. Modern ethylene plants embody a high degree of process interaction and integration with complex supporting equipment networks aimed at achieving these ends. These energy recovery considerations are discussed at length by Ruhemann and Charlesworth(32). Haselden(33) deals with similar aspects for air separation processes.

Thermodynamic analysis of ethylene plants shows reversible separation efficiencies of less than 5% (Ruhemann and Charlesworth). As Haselden points out, if some of the products are required to be liquefied, then a significant proportion of the energy input may be consumed by liquefaction with consequent reduction in expected efficiency. However there still exist significant sources of irreversibility which provide opportunities for improvement. Table 3, taken from Haselden for an air separation plant (a similar low temperature gas separation process), indicates the major sources of irreversibility. The greatest energy usage, in compression, is generally beyond the control of the designer so that attention should be focussed primarily on column and heat exchanger losses. These can be physically interpreted in terms of irreversible degradation of "cold" which necessitate increases in expensive refrigeration requirements.

Table 3

Distribution of Losses in Air Separation

Source	Power Consumption (%)	Loss (%)
Compressor Irreversibility	42	-
Column Irreversibility	20	52
Heat Exchange Irreversibility	9	23
Heat Inleak	7	15
Expansion Valves	2	5
Turbine Irreversibility	2	5
Reversible Separation Work	18	-
	100	100

High temperature heat recovery in the product recovery section is of rather lesser importance as the associated energy costs, e.g., for steam are considerably lower.

Column losses are dependent on the temperature difference existing between overhead and bottoms and for a given column can only be reduced by provision of intermediate reflux. Although such temperature differences may be very high, especially early in the separation sequence where a wide range of components exist, associated capital costs appear to preclude such changes in present plants.

Reduction in exchanger losses is in principle much easier, since it can be achieved by minimizing exchanger temperature driving forces. Thus suitable choice of stream matches for exchange can result in increased cold recovery and reduction in process energy requirements. Additionally stream thermal energy levels may be raised or lowered by compression or expansion (e.g., vapor recompression or flashing) to increase thermodynamic contacting efficiency.

Efficient energy utilization may involve considerable stream interaction both within and between individual processing sections. It is with the synthesis of such energy recovery networks of compressors, expanders and heat exchangers that the present program system applications are concerned.

5.3 Major Process Assumptions

There are a number of assumptions inherent in the application studies. They do not lead to great loss of generality but should be stated at this point. They are as follows:

i) Plant feed and capacity are fixed. No allowance is made for process modification due to feed changes. Nor are the effects of over-design for future expansion or for safety and/or maintenance purposes considered.

ii) Column arrangement and operating conditions are fixed. The configuration of the columns is pre-specified as are individual column operating conditions, i.e., product compositions, operating pressure and reflux ratio. It is convenient to set the latter as the ratio between actual and minimum reflux (R/R_{MIN}). The value is dictated by the economic balance between operating and capital costs. The general value used in this study was 1.2, reducing to 1.1 for particularly low temperature columns where condensation costs are high. The reader is referred to Perry[34] for further details.

iii) Components present in small quantities (CO, CO_2, N_2, etc.) are neglected. All heavy components (C_{4+}) are treated for convenience as n-butane.

iv) Refrigerants are assumed pure. Refrigerant systems are in fact filled from product lines and thus contain some impurities, which will have some effect on evaporation levels but little on circulation rates. However the advantages of reduced computation time are considered to outweigh the loss in generality. Refrigerants are also assumed to transfer only latent heat during use.

v) All process equipment pressure losses are neglected.

5.4 Refrigeration Unit (RUNIT)

i) General

The type of cascade refrigeration unit employed in ethylene plants has a fairly standard configuration (King[23]). Thus there is little point in attempting to synthesize such a unit by starting from basic principles. However, changes in refrigeration demands and levels of availability require frequent re-computation of the unit and make automated computation highly desirable. A routine, RUNIT, has been written to perform this function.

This routine has been programmed as a skeleton flowsheet generator which automatically generates and costs an equipment network for any given set of refrigeration demands. The approach may be regarded as being intermediate between simulation and synthesis. The standard flowsheet generated for a typical unit is shown in Figure 8. It shows two cascaded refrigerant circuits employing propane and ethylene refrigerants with two process levels for each. The process is essentially simple compression refrigeration with the usual compression-condensation-flashing-evaporation steps. Several features are added. The saturated liquid refrigerant may be sub-cooled by contact with one or more process streams to permit recovery of refrigeration with reduction in refrigerant circulation. Especially at the lower temperatures this may extend to completely internal streams as is shown in the ethylene section. Cold is recovered by cross-exchange between the evaporated vapor and the liquid before flashing. Within the multi-stage compression train water intercooling may be employed, where temperatures are high enough, to reduce compression power requirements.

ii) Computational Sequence

Within each refrigerant circuit the computation sequence is as follows. Firstly an iterative sequence is required to determine the refrigerant circulation rates. For this purpose the circuit may be divided into

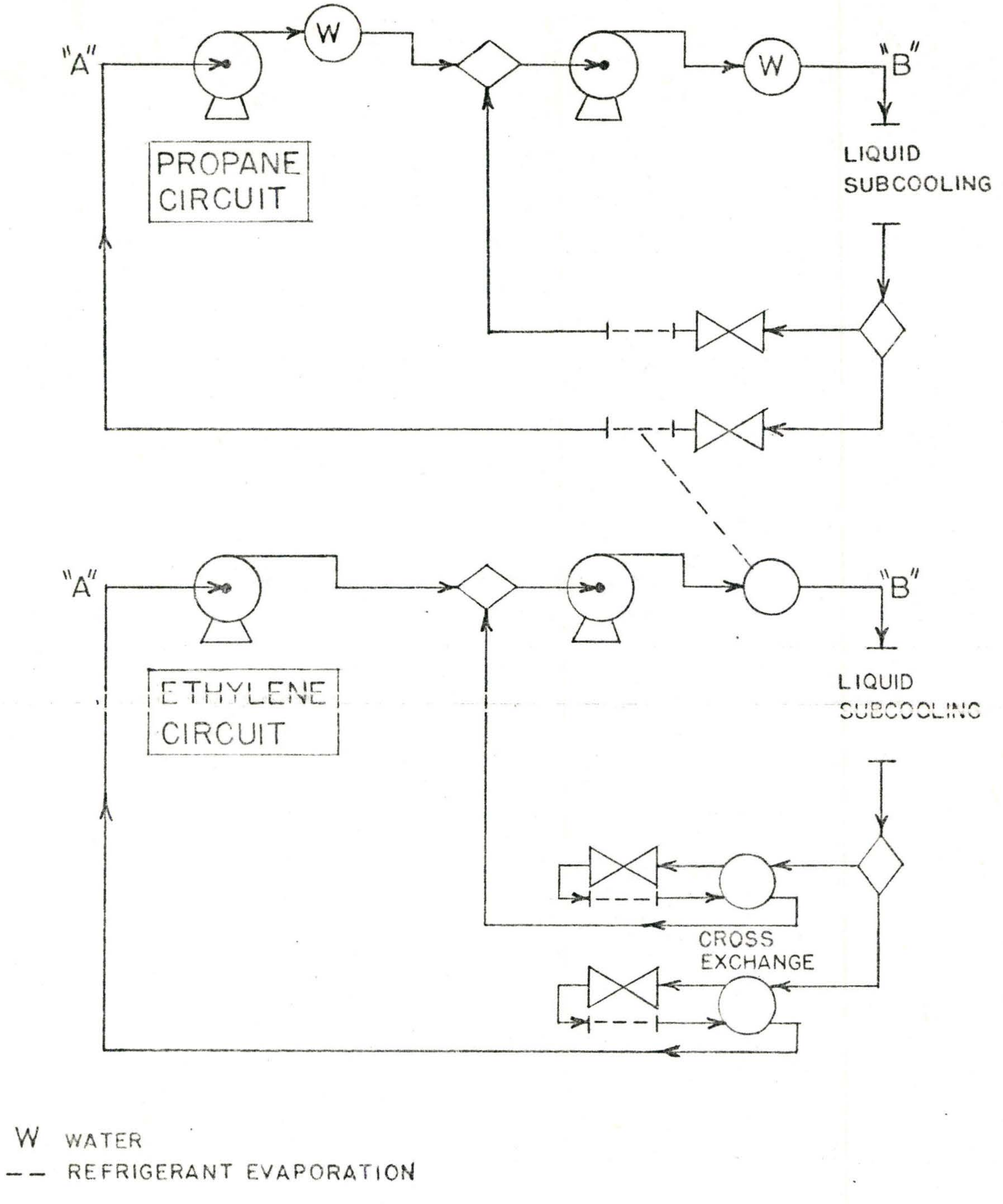


FIGURE 8. TYPICAL CASCADE REFRIGERATION UNIT

two parts; liquid sub-cooling, flashing and evaporation, ("B" to "A" in Figure 8) and compression and condensation, ("A" to "B"). The iteration only involves the former. Starting from "B" where the stream is a saturated liquid, the total flow is estimated from the total circuit refrigeration load. Then the sub-cooling by process streams can be computed to obtain the stream condition prior to flashing down to the individual levels. From these flash calculations the refrigerant flow necessary to satisfy the demand for each level is estimated. Where cross-exchange is used a separate iteration around each flash/cross-exchange loop is required. The total flow is obtained by summing level flows to begin the next iteration.

When the total flow has converged, the compression train-condensation section can be computed directly. The refrigerant circuits in the cascade must be computed in increasing order of temperature since condensation loads for lower circuits must be added to process refrigeration demands for the next highest circuit.

iii) Refrigeration Levels

The selection of refrigerant temperature levels is a difficult problem for which there is little theoretical guidance. The number of levels for each refrigerant can be limited to a maximum of two or three by practical considerations such as minimizing control problems and compressor costs. The spacing of the levels is more difficult. There is some thermodynamic basis (lower energy requirements) for level spacing so as to give approximately equal compression ratios between stages. However the effect of the possible process demand levels must be considered as the selection of levels influences refrigerant costs which in turn influence the refrigerant demands at those levels. A large process demand at a particular temperature may dictate the provision of refrigerant at that level. An "optimum" set of levels for any particular process should exist but would be somewhat difficult and time consuming to establish, especially as both continuous and discrete variables are concerned.

The approach taken in this study was initially to choose approximately equally spaced levels (equal pressure ratios) and then to make some subsequent adjustments for specific process demand levels. It is felt that the final results represent reasonably good and practicably realizable designs.

iv) RUNIT Operating Details

The present refrigeration routine, RUNIT, can handle up to a total of ten process refrigerant levels with up to three arbitrary refrigerant species (three circuits). Those used for the present processes were methane, ethylene and propane. Internal cross-exchange was used only for the two lowest level circuits. A typical computation time on the CDC 6400 was around 5 seconds. A graphical algorithm and further details of RUNIT are given in Appendix II.

CHAPTER 6

HIGH PRESSURE PROCESS

6.1 Process Considerations and Problem Computation

The first application is to a conventional high pressure process as described in section 5.1. The basic separation scheme is shown in Figure 9 and feed details are given in Table 4. The feed composition is typical of plants cracking a propane feedstock with conditions corresponding to those after the acid gas removal step. Feed rates for modern plants may be rather higher than those shown; however the present values approximate those for an existing Canadian ethylene plant which served as a guide for this first application study. Operating conditions for all columns are given in Table 5. Once again product specifications for recently built plants may be rather higher than those shown. Table 5 also gives specifications for the feed and for two product streams from which "cold" may be recovered. These are the demethanizer overhead tail gas and the liquid ethane product stream from the bottom of the C_2 splitter.

The cascade refrigeration unit can be treated conveniently as a separate sub-process. It may "buy" cooling from or "sell" waste heat to the main processing sequence which forms the other sub-process. This decomposition serves the double purpose of reducing problem size and preventing unwanted interaction between streams in the two sub-processes. The configuration of the refrigeration unit is virtually fixed as the only optimization decision to be made is in ordering the use of at most two purchased streams. This is easily carried out by hand so that the full synthesis procedure is to be applied only to the main processing sequence.

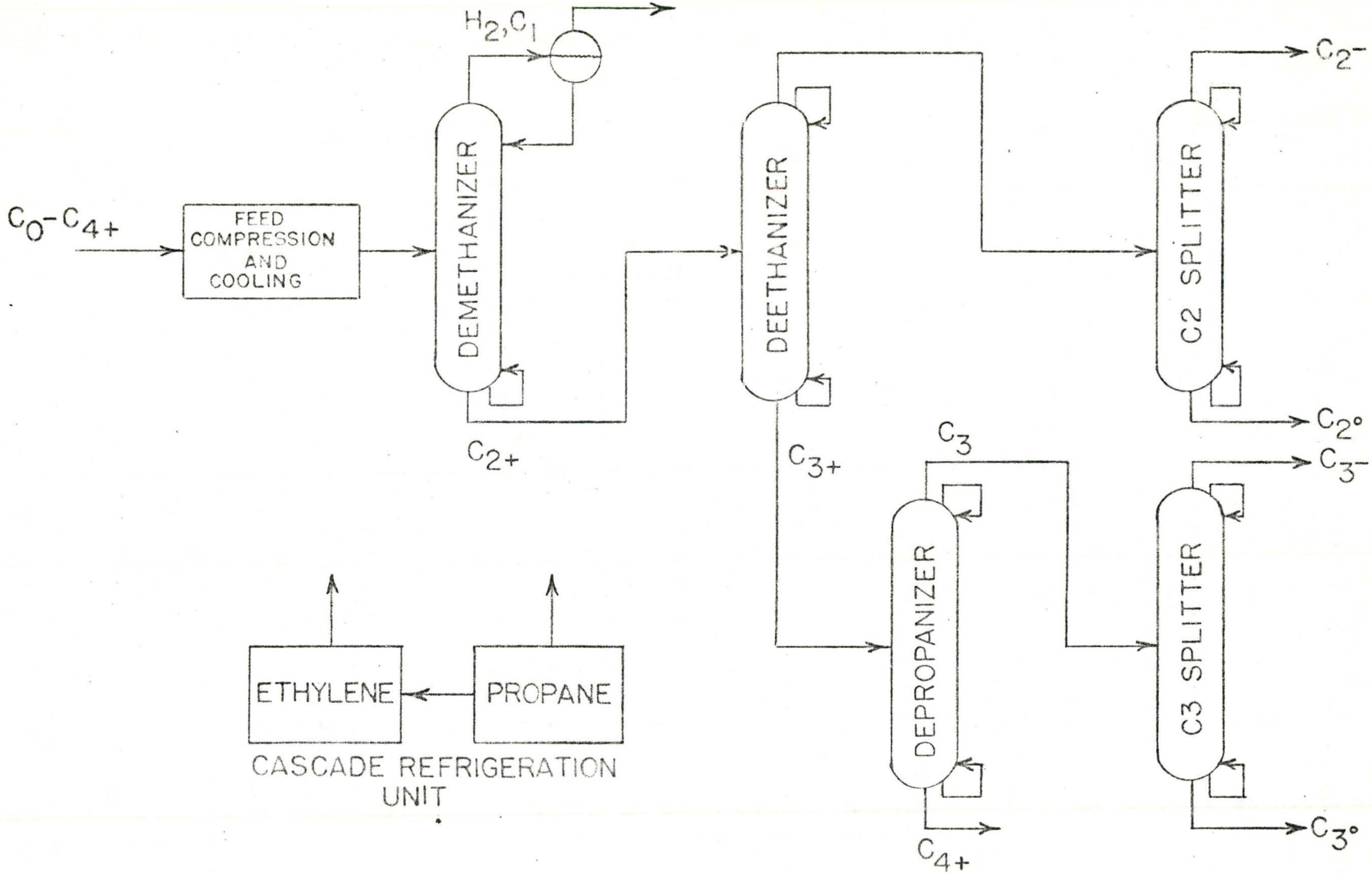


FIGURE 9. HIGH PRESSURE ETHYLENE PLANT - SEPARATION SCHEME

Table 4

Process Feed Details

Composition (Mole %) -

Hydrogen	17
Methane (C_1)	33
Ethylene (C_2-)	21
Ethane (C_2o)	14
Propylene (C_3-)	9
Propane (C_3o)	3
Butane (C_4)	3
	100

Total Flow	1500 lb moles/hr.
Pressure	115 psia
Temperature	$60^{o}F$

Table 5

High Pressure Process Operating Conditions

Column Conditions

Column	Pressure (psia)	R/R_{Min}	Key Splits (Mole fractions)		
			Keys	Overhead	Bottom
Demethanizer	565	1.1	C_1	0.65	0.01
			C_2-	0.01	0.43
Deethanizer	465	1.2	C_2o	0.39	0.015
			C_3-	0.025	0.47
C_2 Splitter*	215	1.2	C_2-	0.96	0.01
			C_2o	0.02	0.93
Depropanizer	200	1.2	C_3o	0.25	0.04
			C_4	0.04	0.84
C_3 Splitter	115	1.2	C_3-	0.90	0.08
			C_3o	0.08	0.76

Additional Stream Specifications

Demethanizer feed temperature	$-60^{\circ}F$
Demethanizer tail gas pressure	215 psia
Ethane product pressure	115 psia

* Overhead product to be drawn off as vapor, not condensed.

Analysis of the column system by COLSYS shows a total of 14 streams requiring further processing. There are 7 hot and 7 cold streams which may be further categorized as follows:

Feeds	1
Intermediates (all reflux)	10
Products	2
Compressed propane vapor	1

This latter propane vapor stream comes from the refrigeration unit. It is a waste heat stream which must be condensed either by cooling water or by its use as a heat source within the process.

Three additional stream matching heuristics are now introduced, all aimed at reducing network complexity and thus minimizing start-up and control problems. They are:

i) Exclude feed/reflux matches.

This helps to ensure reliable and well-controlled reflux generation which is essential for satisfactory column operation.

ii) For reflux streams allow only one process/process match, then satisfy residual by services.

The same considerations apply as for (i).

iii) Set a minimum heat load for process/process exchange.

This helps to avoid a proliferation of residual streams which have only been very slightly processed, a situation which leads to greater network complexity.

A further heuristic was introduced to limit the maximum stream temperature reduction achieved by a single level of refrigerant. It was aimed at conserving low level refrigeration. The value used was 50°F which corresponded approximately to the situation in the actual plant mentioned earlier.

The synthesis system may now be applied to the solution of the problem. The solution sequence is shown in Figure 10. Note the following points.

i) Initial (low temperature) energy costs were established by prior computation of self-standing refrigeration unit examples. Subsequent passes through the refrigeration routine serve to adjust these values.

ii) The solution of each general sub-process problem requires separate computation of the SMATCH-ENERGY-BRBND sequence; the present case involves only one such sub-process so that this sequence is computed only once for each overall iteration.

iii) Re-computation of SMATCH, the stream processing path generating routine, on successive iterations is only necessary when stream flows change. The only such case in the present example is the compressed propane vapor from the refrigeration unit, whose flow changes with refrigeration demands and cold stream sales.

iv) The overall computation sequence is converged when there is no change in configuration, and thus process cost, between successive iterations. This is attained when energy costs have been established within the correct range to render all sub-processes truly independent.

v) The computation sequence shown in Figure 10 must be repeated for each new set of discount parameters. Since the present example involves only one discount parameter a simple trial and error scheme was used to determine the effect of the parameter on optimal network cost and configuration (refer to section 6.4).

The configurations of the optimal process network and its associated refrigeration unit are shown in Figures 11 and 12. Note that in Figure 11, hot streams are denoted by positive numbers and cold streams by negative numbers.

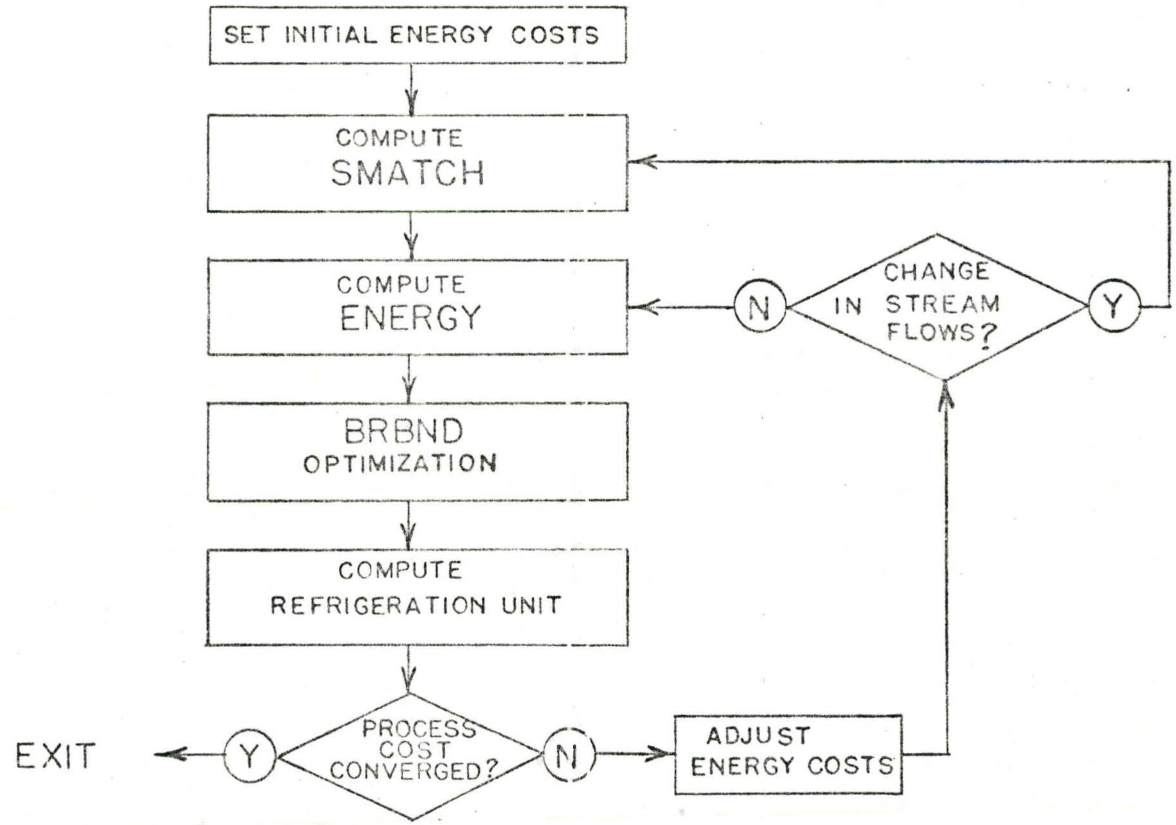


FIGURE 10. PROBLEM SOLUTION SEQUENCE

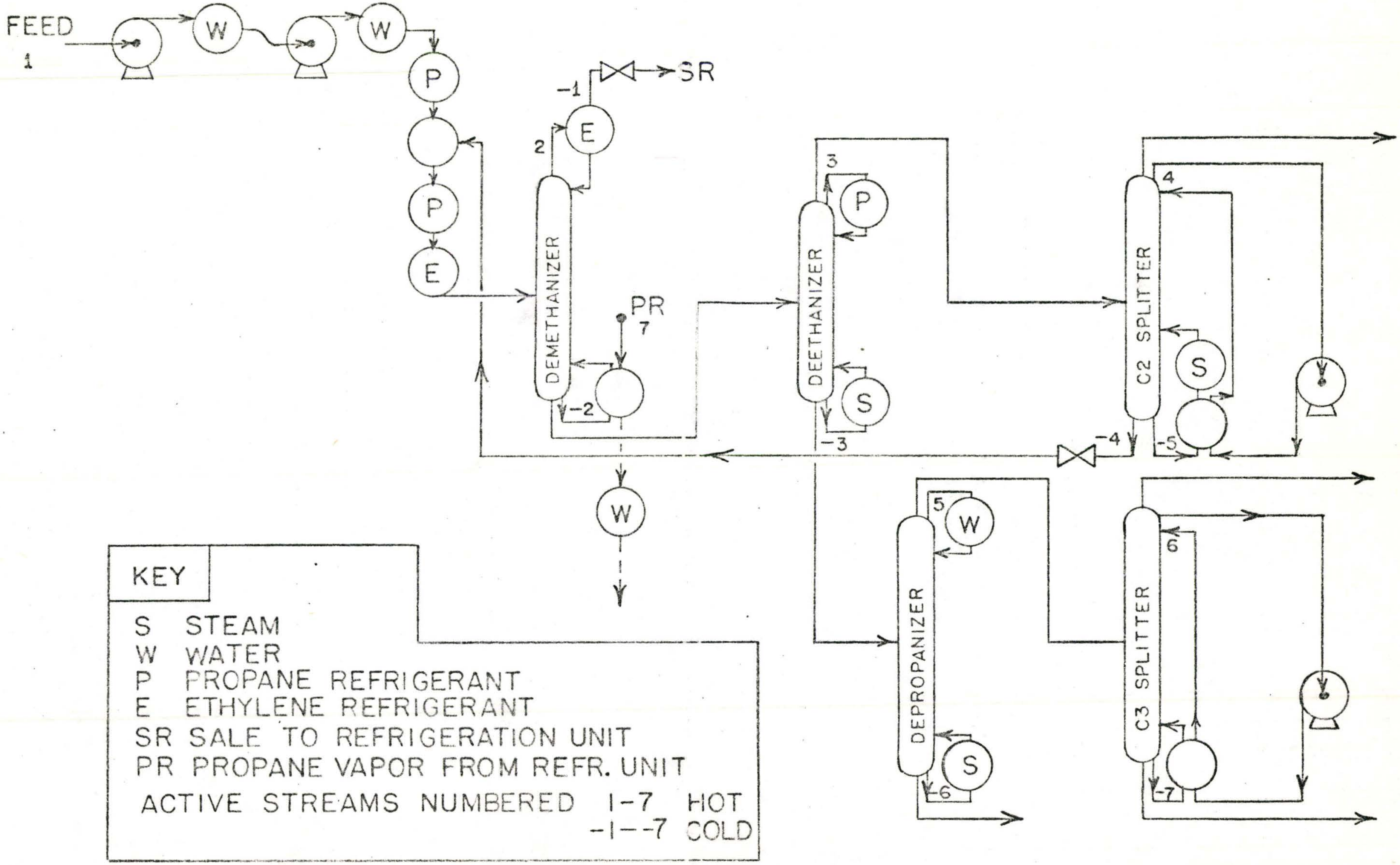


FIGURE 11. OPTIMAL PROCESS CONFIGURATION (H.P.)

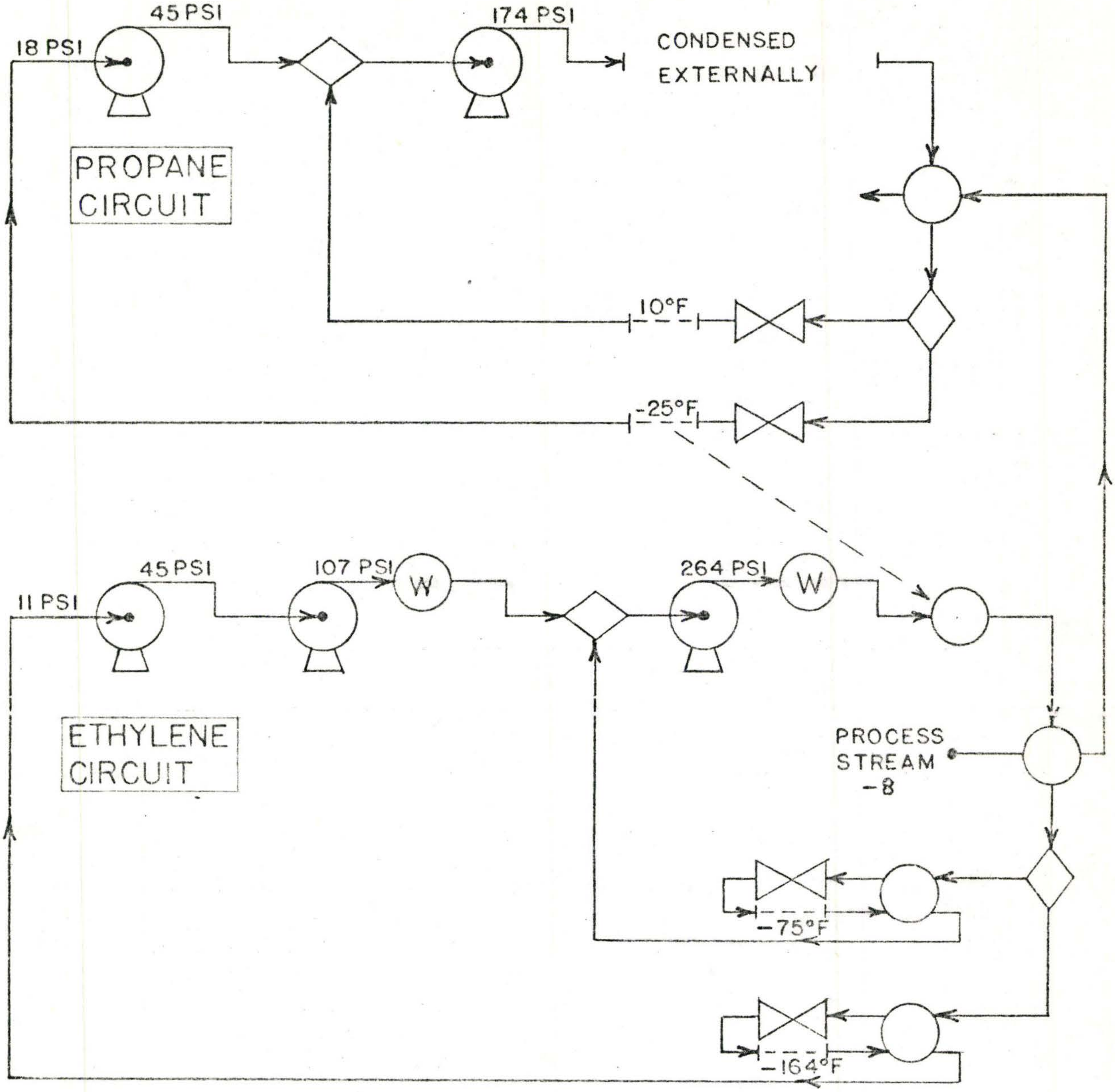


FIGURE 12. REFRIGERATION UNIT FOR HIGH PRESSURE PROCESS

Complete process details are given in Appendix III.1. The optimal configurations are found to be essentially identical to those for the existing plant mentioned earlier. The process requires a comparatively low degree of interaction between process streams and thus is satisfactory from a start-up and control standpoint. An interesting point is that in spite of the ready availability of waste heat both C_2 and C_3 splitter columns are reboiled by vapor recompression. This is primarily dictated by the very high refrigeration costs for any alternative means of overhead condensation.

6.2 Entropy Aspects

In the light of the results the heuristic with the most interesting effect is that which limits the entropy increase for any process/process stream exchanger match. It was designed to conserve refrigeration by minimizing total process heat exchanger irreversibilities. It acts by either i) preventing stream matches which contribute too largely to process irreversibility or by ii) effectively forcing such matches to be delayed until one of the streams has been processed to the extent that a match can be made with a tolerable degree of irreversibility. Thus the maximum entropy increase parameter (DENMX) affects both the pattern and sequencing of stream matches. It was found to have a significant effect on the size of the problem to be solved and relevant solution statistics for the SMATCH and BRBND computations are shown in Table 6 for three parameter values. Times quoted are for a CDC 6400 computer for three levels of branching. A DENMX value of 25 corresponds to constant temperature heat transfer at a temperature of around 50°F with a temperature difference of around 60°F. The value can be estimated from the approximate relation

$$\mathrm{DENMX} \doteq \frac{a}{T^2} \qquad (15)$$

Table 6

Effect of Maximum Entropy Change Parameter (DENMX) on Problem Size

DENMX ($X10^5/^oR$)	Total No. of Streams	No. of Equipments	No. of Processing Paths	SMATCH Time (seconds)	No. of Path Combinations	Max. Size Sorting Problem	BRBND Time (seconds)
25	40	53	37	10	$0.18 * 10^6$	34	7
30	42	57	40	12	$0.58 * 10^6$	22	10
40	47	66	48	18	$3.20 * 10^6$	78	15

where a is the temperature difference and T the average (absolute) exchange temperature. The derivation is given in Appendix I.2.

This temperature function clearly shows the correct trend with respect to low temperature energy usage as it strongly reflects the increasing energy value with decreasing temperature.

It can be seen from Table 6 that the total problem size increases rapidly with DENMX, although the corresponding branch and bound solution time increases rather more slowly. Examination of the optimal network shows that the maximum entropy increase value for both process/process and process/refrigerant exchange is almost exactly 25. It is interesting to note that for both vapor recompression exchangers the values are below 8.0; this may partially explain the apparent desirability of vapor recompression.

It may not always be easy to establish a priori a suitable value for the upper limit, DENMX, which does not allow the possibility of excluding an exchange match which forms part of the optimal network. However the value of such a heuristic is evident especially as it is so simply implemented and has the advantage of some theoretical thermodynamic basis.

6.3 Branching Problem Selection

Implications of the branch and bound algorithm can now be examined. The efficiency of the branch and bound procedure is most usefully measured by the sizes of the problems to be solved at the final level of branching. This can be illustrated by examining a typical distribution of calculation times for the whole branch and bound procedure, given below:

Establishing initial good upper bound	25%
Executing general branch and bound logic	25%
Solution of final level problems	50%

The final entry is strongly dependent on the size of problems; in particular the time taken to sort the network costs is a very non-linear function of problem size. Another related consideration which may also be important is the amount of core storage required to solve large size problems.

As is seen in section 2.1, the branch and bound concept makes no restriction on how to choose bounding problems as long as the problem set at each level mutually bounds the original problem. However bounding problem choice has been found to have a strong influence on resulting problem size and especially for automated solution the need for an efficient, systematic selection method is obvious. The major consideration is to select problems so that many processing paths are eliminated for each problem at each level,thus achieving considerable reduction in problem size. This should not however be achieved at the expense of creating too large a number of problems.

A very satisfactory set of rules for problem selection for the two process examples considered in this study is given below. Problems are classified according to the stream match on which they are based. Streams are described by the number of exchange steps they have undergone, e.g., a secondary stream has been processed by one exchanger [P = Primary, S = Secondary, T = Tertiary].

Branching Level	Problem Type (Stream Match)
1	P/P matches
2	P/S and S/S matches
3	P/T, S/T and T/T matches

This set has two convenient computational advantages. Firstly the problems are conveniently located in the first, second and third equipment rows respectively of the processing path array. Secondly since the problem subsets for each level are mutually exclusive, a minimum of checking is required to avoid duplication of problems.

There are two important restrictions.

i) Problems at any level which do not lead to elimination of a sufficient number of paths will not produce a satisfactory reduction in problem size and are consequently disregarded.

ii) There is a minimum number of bounding problems branched from any node (refer to Figure 1). This is because the final problem in the set, which excludes all paths containing any of the bounding problem matches for other members of the set, will otherwise produce too few eliminations. This will lead to the same difficulty as in (i). This minimum number is always achieved by adding problems of the level 1 type to those lower level branchings which fail to meet the minimum.

Restrictions (i) and (ii) give rise to two adjustable parameters, in addition to the number of branching levels, which may be chosen to give best results for any particular problem.

Process/service matches were found to produce an insufficient number of path eliminations to be useful.

The value of establishing a good initial feasible network (whose cost forms an upper bound on subsequent network cost) in rejecting paths which must lead to higher cost networks was found to be considerable. This was in spite of the somewhat involved and time-consuming program logic which was found necessary to establish it. The reader is referred to Lee et al.[10] for further details.

6.4 Stream Pricing and Exergy Considerations

i) Process Decomposition - Discount Parameter

The trial and error price adjustment scheme and its implementation by use of a discount parameter have been discussed earlier in section 3.3. The results of its application to the present process should now be considered. For this process there are two categories of stream transfers.

The first involves the sale of refrigerant propane vapor to the main processing sequence. This is however a sale which is already fixed, i.e., it was decided a priori to process this stream completely within the column system, condensing as much as possible by its use as a heat source for process matches and water condensing the remainder. Thus the configuration of its use is an internal optimization decision for the column system and no transfer price need be assigned.

The other category of sale is that of cold process streams to the refrigeration unit for cold recovery. This situation does demand the assignment of transfer prices in order to be able to decide which combination of streams are to be sold. The streams involved are numbers -1 and -4 and their residuals. The transfer price function, based on the cold section cost spline, is to be adjusted by a single discount parameter, δ. Thus the overall decomposition problem for this process can be expressed in terms of only one price adjustment variable.

An estimate of the value of this δ parameter may be obtained by physical reasoning, as follows. For this process, particularly in the low temperature region, energy costs are considerably greater than equipment costs. Hence from physical considerations δ may be expected to reflect primarily the relative degree of degradation of cold between internal use and use within the refrigeration unit. For internal use of any cold stream there is a single degradation step in its use for process/process exchange; if such a stream is sold to the refrigeration unit there are two degradation steps involved before useful process cooling is produced. The first is in exchange for cold recovery within the refrigeration unit and the second is in the process/service exchange involving the use of the refrigerant which may be regarded as being produced as a result of the cold recovery step. Thus δ can be expected to reflect this single extra degradation step and thus be of the order of the minimum exchanger approach temperature (ΔT_{Min}). This is shown to be the case in Figure 13 where the optimal process cost is shown as a function of δ (expressed as a multiple of ΔT_{Min}). It can be seen that any positive value of δ leads to the overall optimum.

In this simple case there are in fact only three possible combinations of stream sales produced by different configurations of use of stream -4. Since all of these cases have been evaluated (Figure 13) it can be guaranteed that the overall optimum has been found. The fractional cost margins between the three appear to be small but it should be pointed out that the high proportion of invariant costs in the overall yearly figure somewhat dampens the real value of the improvement produced.

ii) Exergy in Stream Pricing

The final energy value splines, which it will be remembered are obtained from service costs, are shown in Figure 14. The figure also shows the exergy or availability function suggested by Tribus and Evans[18] and described in section 3.3. The correspondence between

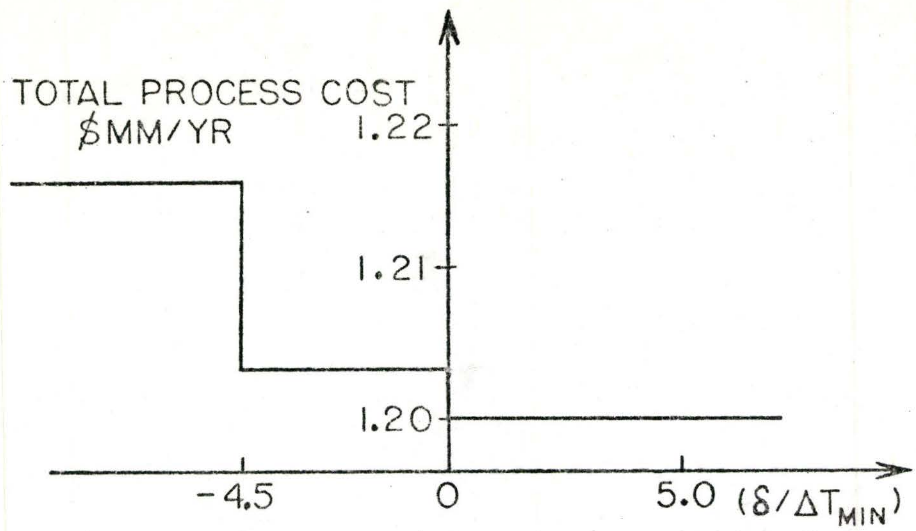


FIGURE 13. EFFECT OF "DISCOUNT" PARAMETER ON OPTIMAL PROCESS COST

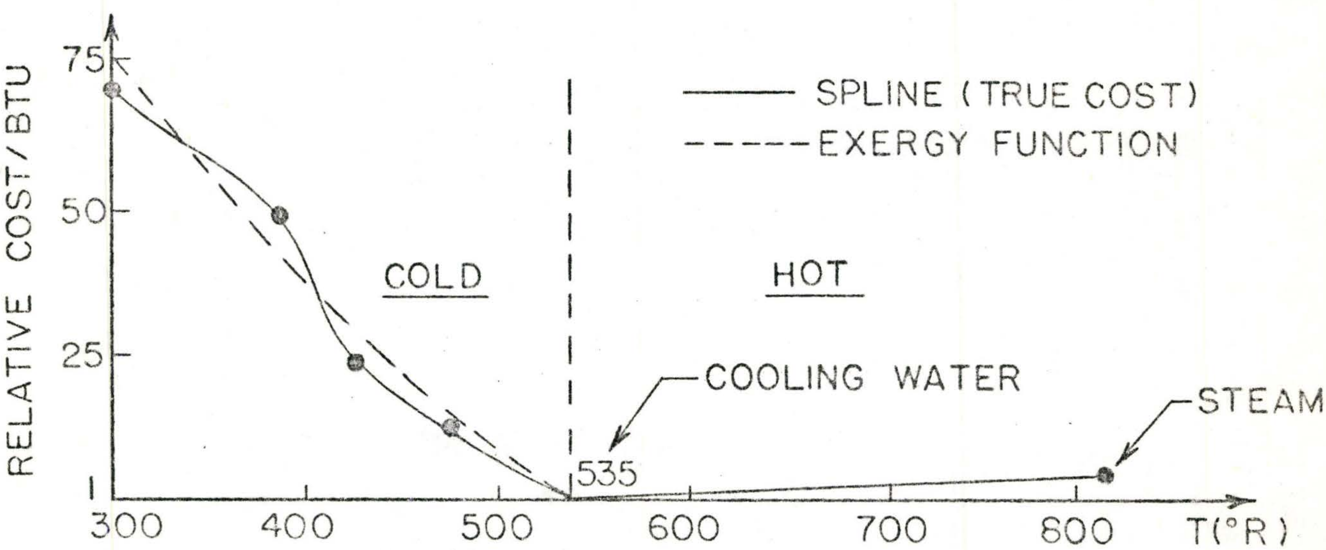


FIGURE 14. ENERGY COST RELATIONS

that function and the cold section spline is seen to be reasonably good. This suggests that the exergy concept may prove a useful one with regard to cold recovery as well as for the high temperature region with which Tribus and Evans were concerned. Although not needed in the present study where physical cost figures were readily available, the simple form of the exergy function may be useful for interpolation and/or extrapolation of energy values in other less well defined situations.

iii) Exergy in Equipment Driving Forces

The exergy concept may also be used qualitatively to examine equipment driving forces. This applies in particular to the heat exchanger minimum approach temperature which may determine the average thermal driving force for exchange. Tribus and Evans[18] have shown that the optimal equipment driving force can be related to the ratio of equipment to exergy costs, i.e., the lower the cost ratio the lower the desirable approach temperature. For the present process, particularly in the low temperature region, equipment costs are very low compared with those for exergy; the ratio of cost for exchangers using refrigeration to that for refrigerants is around 10:1. Thus a very close approach temperature with consequent reduction in exchanger irreversibility is desirable.

This can be illustrated with respect to the refrigeration unit. Although a more usual design figure for the approach temperature is $15^{o}F$, a reduction to $10^{o}F$ gave a total unit cost saving of around 4%. This was due mainly to a decreased propane circulation resulting from a lower condensation temperature/pressure. A further reduction in condenser driving force results from reducing the cooling water temperature rise from $20^{\circ}F$ to $10^{\circ}F$; this produced a further cost saving of around 8%. Thus a value of $10^{\circ}F$ was used for both of these parameters (with one exception, described later) throughout the study. Some further decrease may be theoretically desirable but, particularly for the approach temperature, may be subject to equipment limitations.

CHAPTER 7

LOW PRESSURE PROCESS

7.1 Process Considerations and Problem Strategy

The second application is to a low pressure ethylene separation process which involves a somewhat different separation sequence from the previous example. The basic separation scheme is that described by Baldus and Linde[30] and shown in Figure 15. There are several features which require comment. The first separation step is deethanization rather than demethanization and requires feed compression to 250 psia as against 565 psia for the high pressure process. The C_{3+} bottom stream is processed in the same sequence as in the high pressure plant. It is in the separation of the C_{2-} overhead stream that the major differences are found. This stream is first cooled low enough to condense virtually all C_2s while still leaving essentially all of the hydrogen and some methane in the vapor phase. These light components can then be removed without fractionation. This decreases the load on the demethanizer which operates conventionally except for the manner of overhead reflux generation. Since its overhead product is essentially pure methane, reflux is produced by feeding the overhead directly into the methane refrigeration circuit. C_2 splitting is achieved by a double column system similar to the configuration used in air separation. Due to the smaller temperature differences between the column ends the system is potentially more efficient than a single column.

The much lower separation pressures for this process (250 psia down to 20 psia) result in much lower temperatures than for the high pressure process and this requires the addition of a methane refrigeration circuit. Advantages claimed for the process over high pressure operation include lower

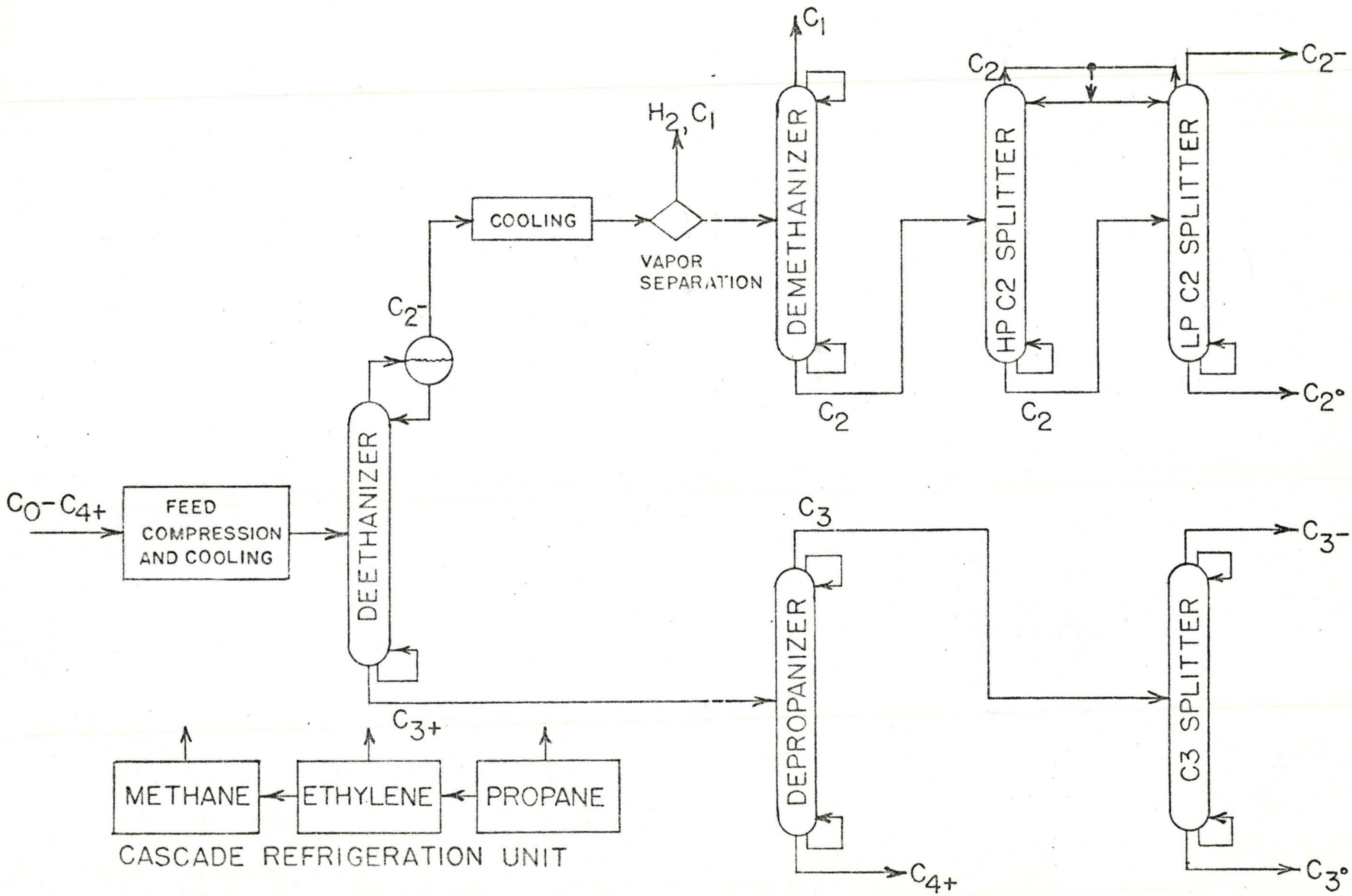


FIGURE 15. LOW PRESSURE PLANT – SEPARATION SCHEME

overall power consumption and easier separations, both due to the lower operating pressures. An increase in thermodynamic efficiency or effectiveness of cold recovery is also claimed.

The feed conditions are the same as for the high pressure process as are the product specifications. It should be noted that this process achieves slightly higher ethylene recovery than the former and this is discussed later. Operating conditions for all columns are given in Table 7 which also gives necessary additional specifications for feed, intermediate and product streams.

As before, the refrigeration unit is treated as a separate sub-process. However for this process, examination also shows that the main processing sequence can be conveniently divided, on a temperature basis, into two non-interacting sections thus further reducing problem sizes. This division is somewhat arbitrary but on examination is seen to be quite logical and justified in terms of the reduction in problem complexity and minimization of unwanted stream interactions.

The <u>first</u> section is comprised of the deethanizer, depropanizer and C_3 splitter and covers a temperature range of 160^oF down to -64^oF. There are no streams suitable for recovery of cold by sale to the refrigeration unit.

The <u>second</u>, low temperature, section consists of the demethanizer and high and low pressure C_2 splitters and covers a temperature range of -64^oF down to -210^oF. Two product streams are available for cold recovery and since both are vapor and thus not storable, an a priori decision was made to sell both to the refrigeration unit to make for easier start up and control. At a later stage (refer to section 7.4) a liquid stream was made available for cold recovery; again an a priori decision was made to sell this stream to the

Table 7

Low Pressure Process Operating Conditions

Column Conditions

Column	Pressure (psia)	R/R_{Min}	Key Splits (Mole Fractions) Keys	Overhead	Bottom
Deethanizer	250	1.2	C_2o	0.16	0.003
			C_3-	0.015	0.57
Demethanizer	75	1.1	C_1	0.98	0.005
			C_2-	0.01	0.58
High Pressure C_2 Splitter	65	1.1	C_2-	0.96	0.03
			C_2o	0.45	0.54
Low Pressure C_2 Splitter*	20	1.1	C_2-	0.96	0.04
			C_2o	0.01	0.94
Depropanizer	200	1.2	C_3o	0.25	0.04
			C_4	0.04	0.84
C_3 Splitter	115	1.2	C_3-	0.90	0.08
			C_3o	0.08	0.76

Additional Stream Specifications

Deethanizer feed temperature $-40^{\circ}F$

Demethanizer feed temperature $-190^{\circ}F$

* Overhead product to be drawn off as vapor, not condensed

refrigeration unit. Since this removes any decision making regarding use of such streams there is no sub-process integration problem in this process application.

7.2 Pseudo-Service Stream Usage

Before proceeding to the solution of the problems a change is made in the method of handling the compressed propane waste heat stream from the refrigeration unit. For the high pressure process this was series processed in the normal fashion. However it could be seen that this stream alone contributed a factor of over 20 to the total problem size. This was largely due to its very large heat content relative to other process streams, which resulted in the generation of a large number of processing paths with changing sequences but essentially identical costs and results. For the high pressure process this stream must be series processed in order to provide sufficient heat at a high enough level to reboil the demethanizer. However it is obviously desirable to avoid this problem of heat content imbalance. This can be done for this external stream (internal streams cannot be treated in this fashion) by providing for parallel usage, designating it as a "pseudo-service" as was done for this application.

Such pseudo-service streams are treated in the following fashion. For any match involving a pseudo-service stream the quantity of it required to satisfy requirements for the other contacting stream is calculated and entered in the exchanger equipment vector. These streams are disregarded by the branch and bound algorithm as their use is always feasible unless the cumulative demand exceeds the supply. Although this difficulty was not encountered in this study, it can be avoided by setting a sufficiently high stream transfer price.

For the propane stream concerned, any quantity not required for pseudo-service usage was water-condensed within the refrigeration unit.

Strictly a transfer price should have been assigned but for the reasons given above this was not required. Thus the effective transfer price was zero; a realistic value would perhaps even have been negative since this pseudo-service waste heat usage would usually lead to a joint saving in both steam and cooling water.

7.3 Problem Computation

Analysis of the first sub-process, the medium temperature section, produces 4 hot and 3 cold streams requiring further processing. There is also the propane pseudo-service stream from the refrigeration unit to be considered. The solution of the problem is handled almost identically to that for the high pressure process and presents no difficulties. The total number of possible networks is only 16 so that the optimal network could readily have been selected by hand. As may be expected the process configuration for the depropanizer and C_3 splitter streams is unchanged as seen in sub-process 1 in Figure 16. Equipment and stream details are given in Appendix III.2.

The low temperature sub-process requires some changes in approach. Computation of the column network identified 3 hot and 6 cold streams available for or requiring further processing. Some rearrangement of these streams was made before proceeding with the solution. The vapor stream pre-separated from the demethanizer feed was combined with the demethanizer overhead product to produce a cold tail gas stream at demethanizer pressure. This was to be sold to the refrigeration unit for cold recovery. The bottom reflux and product streams from the low pressure C_2 splitter were combined into a single stream to be reboiled while providing process cooling. This was to allow comparison with the process configuration described by Baldus and Linde[30]. A separate stream was created from the low pressure C_2 splitter bottom product after reboiling. This was again to be sold to the refrigeration unit. This left 3 hot and 5 cold streams to be processed along with the propane pseudo-service. Since process temperatures were uniformly low it was decided to rule out the use of steam as a service heat source and to replace it with the pseudo-service.

Disregarding the two sold streams and the "hot" overhead stream from the low pressure C_2 splitter (it is actually colder than any remaining cold stream) there remain 5 process streams within a temperature range of $-66^{\circ}F$ to $-117^{\circ}F$, a difference of only $51^{\circ}F$. The heuristic which limits the maximum exchanger entropy increase was found to be ineffective within this narrow temperature range and was thus discarded. (The maximum value for process/process exchange found in the optimal network was 17.3×10^{-5}, cf. 25.0×10^{-5} for the high pressure process.) Due to the very low temperatures and consequent necessity for efficient cold recovery, the first two additional stream matching heuristics introduced for the high pressure application were discarded. They were:

i) Exclude feed/reflux matches

ii) For reflux streams allow only one process/process match, then satisfy residual by services.

This was aimed at permitting greater flexibility in stream matching while hopefully not producing serious process control problems. The heuristics for minimum process/process heat load and maximum stream temperature reduction by a single refrigerant level were retained. The necessity for efficient cold recovery also dictated reduction in the minimum exchanger approach temperature. It was reduced from $10^{\circ}F$ to $7^{\circ}F$.

In spite of the small number of streams involved, the low temperature section problem could not be solved within the existing system data storage capacity. It is estimated that the total problem size (number of possible networks) would have reached around 10^8. In an effort to overcome this difficulty, attention was directed at the processing of the feed stream, 2, which must be cooled from -66°F to -190°F. It became obvious that unless either of cold streams -1, at -87°F, or -2, at -81°F, was initially contacted with the feed, after any one match the feed would be below their temperature range and thus valuable cooling/reboiling would be lost. Hence it was concluded that the optimal configuration must include either the 2/-1 or 2/-2 match. This in effect created two parallel sub-problems one of which would contain the overall optimum. It actually represents a limited introduction of the branch and bound branching strategy at the stream matching rather than optimization stage.

The resulting sizes of the two sub-problems were found to be 1944 and 3024, striking reductions from 10^8. Thus it is seen that an appropriate branching strategy introduced at this stage is particularly effective in problem size reduction.

The configurations of the optimal process network and refrigeration unit are shown in Figures 16 and 17. (The low temperature section is sub-process 2.) Full details are given in Appendix III.2. The optimal solution is seen to contain the 2/-1 match; the cost for the best solution containing the 2/-2 match was substantially higher. The optimal process configuration appears to present no real operating problems, based as it is on conventional vapor recompression cycles for each of the two C_2 splitter columns. The low temperature section configuration may be compared with that described by Baldus and Linde[30], shown in Figure 18. At least for the present feed it is found to cost approximately an additional \$30,000/year.

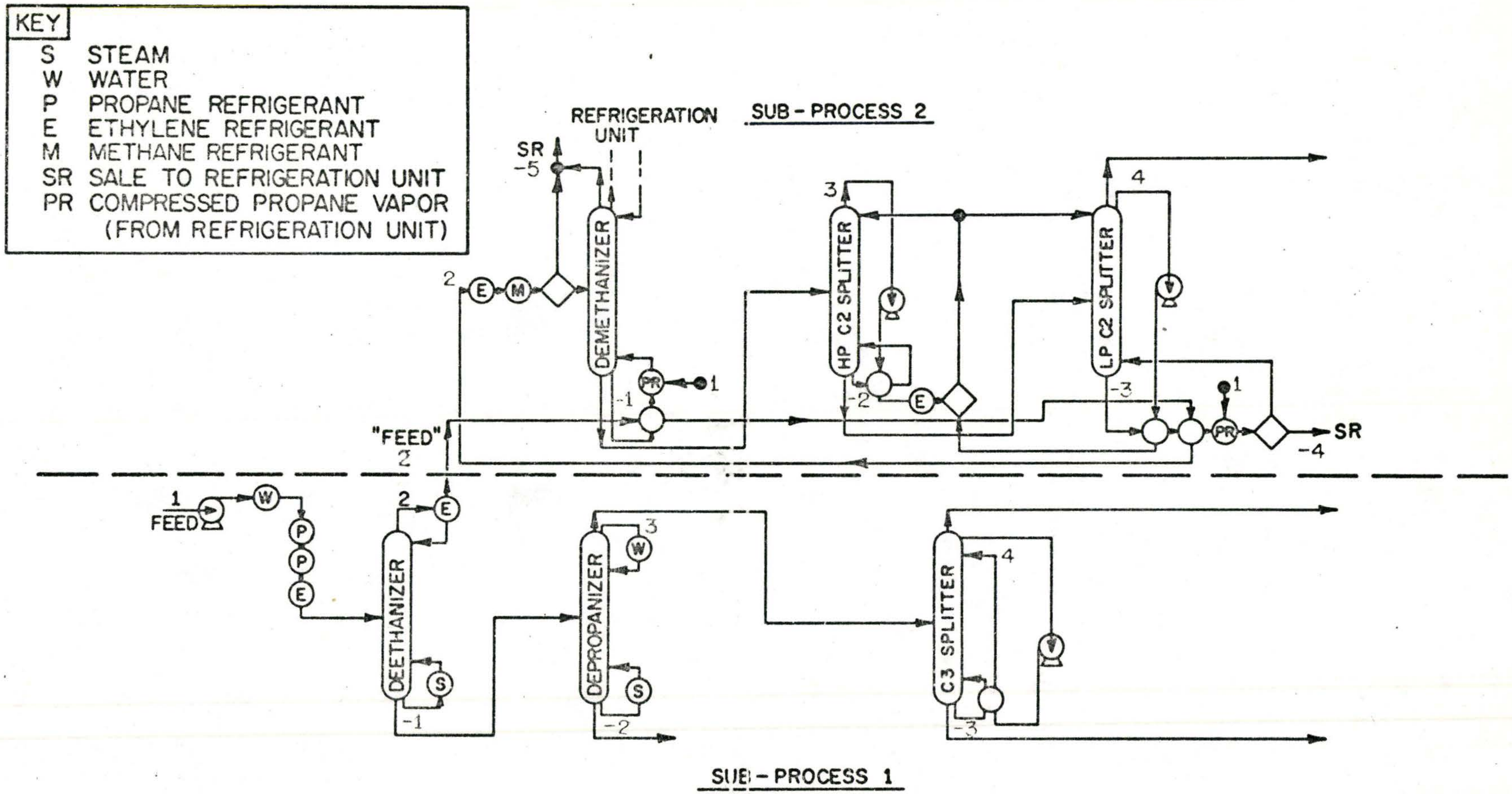


FIGURE 16. CONFIGURATION OF OPTIMAL LOW PRESSURE PROCESS NETWORK

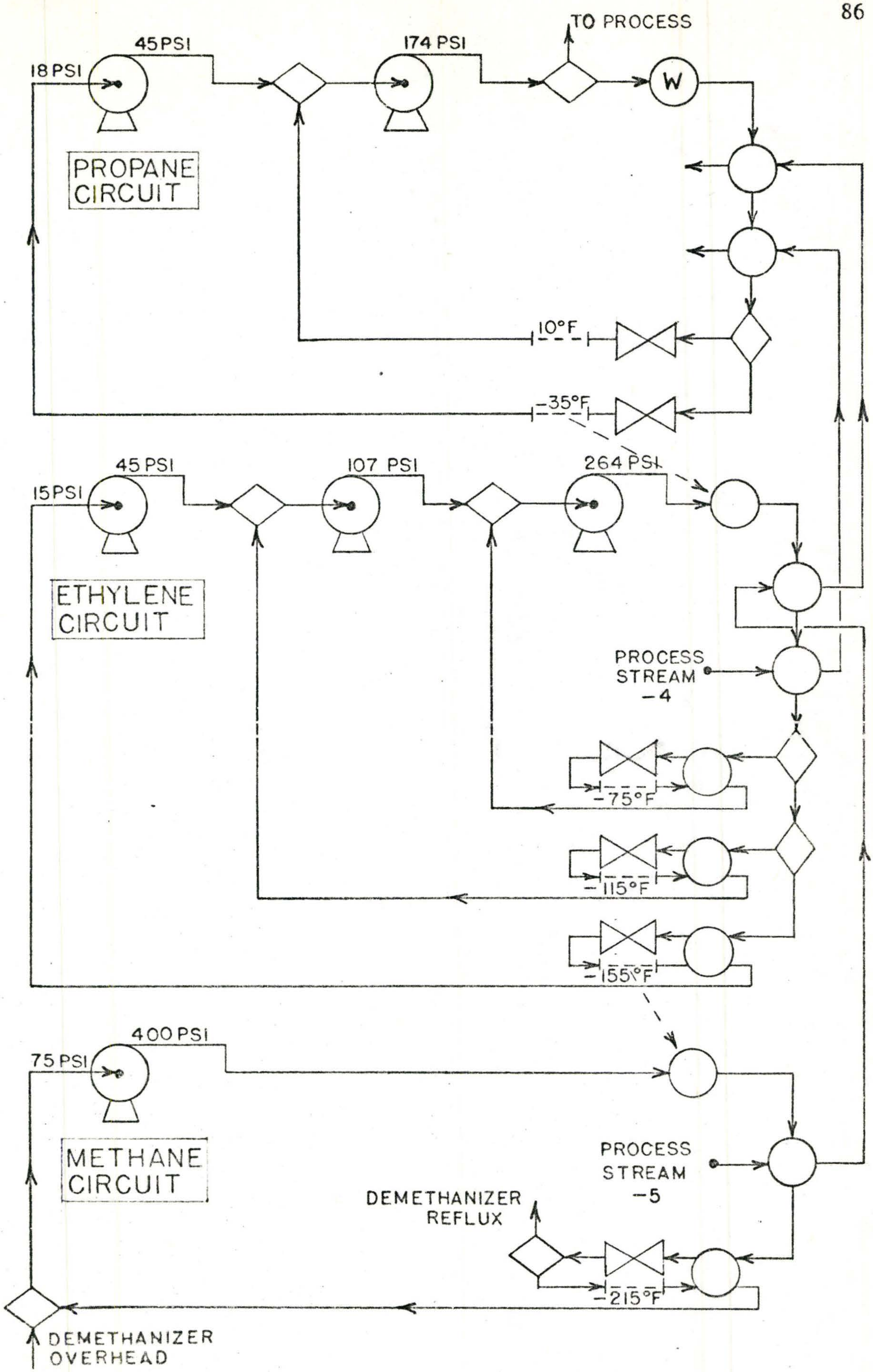
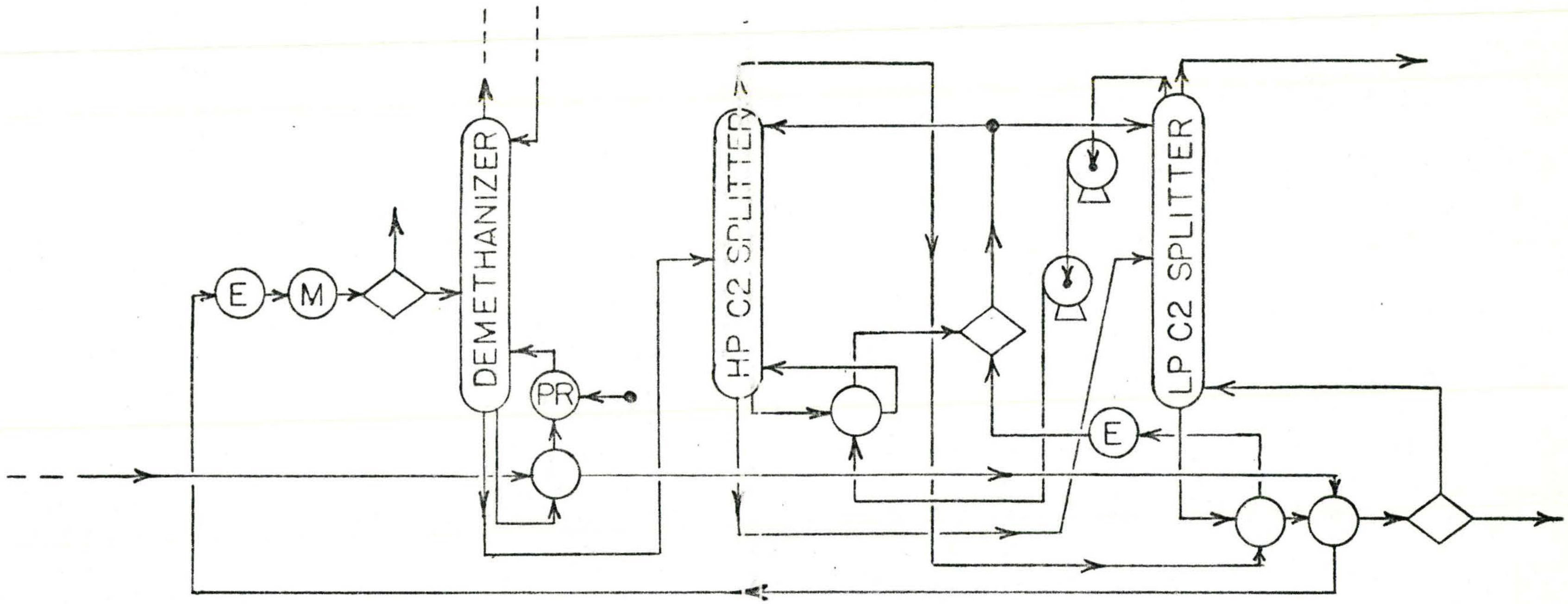


FIGURE 17. REFRIGERATION UNIT FOR LOW PRESSURE PROCESS



KEY

E ETHYLENE REFRIGERANT
M METHANE REFRIGERANT
PR PROPANE VAPOR (FROM REFR. UNIT)

FIGURE 18. PROCESS CONFIGURATION DESCRIBED BY BALDUS AND LINDE

7.4 Modifications to Optimal Process Configuration

Further improvements to the process configuration shown in Figure 16 appear to be possible. For the first consider the means of generating the deethanizer overhead reflux. In the configuration shown this reflux is wholly supplied by partial condensation of the column overhead stream with medium pressure ethylene refrigerant. At the same time only part of the demethanizer bottom reflux can be used to further condense this overhead stream due to the minimum approach temperature limitation. If however the ethylene refrigerant flow is reduced to the point where the whole of the demethanizer bottom reflux stream can be used for condensation, a saving of $73,000/year would result. The only other process change required is the elimination of the use of the propane pseudo-service in providing demethanizer bottom reflux. Of course a portion of the deethanizer overhead reflux would then have to be withdrawn at some intermediate stage from the demethanizer reboiler. This feedback may introduce some problems in start-up and control but the financial incentive to solve them is obvious.

A second possibility should also be investigated. As will be remembered the low pressure C_2 splitter bottom reflux and product streams were combined into a single stream and an additional product vapor stream was created for sale to the refrigeration unit (refer to section 7.3). However it is seen, in Figure 16, that a substantial proportion of this combined C_2 splitter bottom stream is reboiled with the waste heat propane pseudo-service whereas the refrigerant potential of this stream could be more valuably employed elsewhere. In fact the quantity of cold used by the pseudo-service is almost exactly equal to the latent heat of the bottom product portion of the combined stream. Thus it was decided to run another case without combining the two bottom streams, leaving the reflux stream free for internal exchange and the liquid product stream to be sold to the refrigeration unit. As expected the optimal network configuration is found to be essentially unchanged except for the removal of the propane pseudo-service exchanger and the increase in quantity of cold sold to the refrigeration unit. The saving amounts to a very substantial $106,000/year.

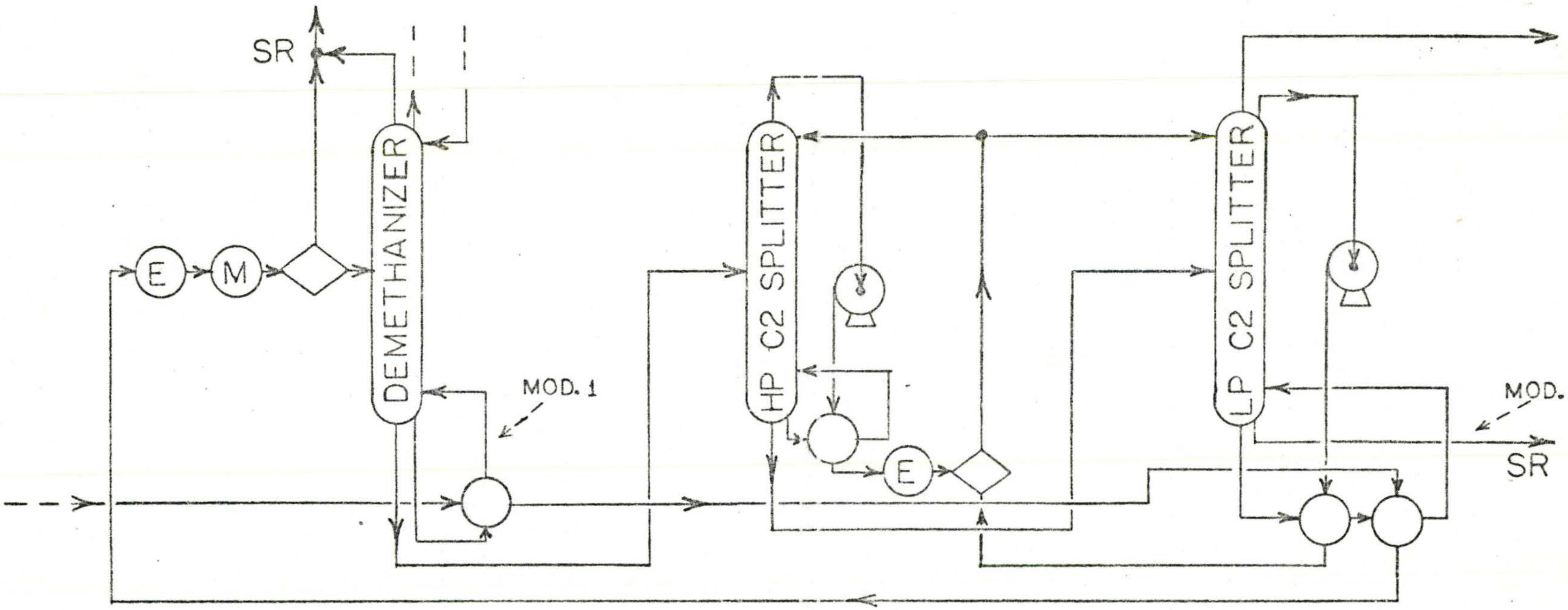
Both modifications are indicated on the modified low temperature section configuration shown in Figure 19.

7.5 Comparison Between High and Low Pressure Processes

A comparison is made between the high and low pressure processes in Table 8. Four cases are presented for the low pressure process, the original solution, then those for the first and second modifications alone and finally that for both modifications. It is seen that for the present feed even the best low pressure case has a substantially higher cost than for the high pressure process. There are several reasons for this, as follows.

i) Comparing the power requirement figures it is evident that the low pressure process never attains the same thermodynamic efficiency as does the high pressure process. The reduction in feed compression requirements for low pressure operation appears to be more than balanced by the increased power requirements for refrigeration.

ii) Further examination shows that a high proportion of total process costs are associated with large cooling loads for feed condensation. The high pressure process requires cooling of only one such stream, while to achieve the much lower temperatures for low pressure operation this must be done for the "feeds" to both sub-processes. In spite of the complex energy exchange networks employed,not all of this cold



KEY

E ETHYLENE REFRIGERANT
M METHANE REFRIGERANT
SR SALE TO REFR. UNIT

FIGURE 19. MODIFIED LOW TEMPERATURE SECTION

Table 8

Comparison between High and Low Pressure Processes(All Costs in $/Year)

ITEM	PROCESS				
	High Pressure	Low Pressure			
		#1	#2	#3	#4
Process Section Costs					
Process Network (Including Columns)	734,000	568,000	568,000	567,000	567,000
Refrigeration Unit	476,000	948,000	874,000	843,000	789,000
Equipment Cost Breakdowns					
Columns	175,000	156,000	156,000	156,000	156,000
Heat Exchangers	231,000	322,000	319,000	315,000	310,000
Compressors	511,000	647,000	604,000	585,000	555,000
Service Costs					
Steam	57,500	67,500	67,500	67,500	67,500
Water	22,200	37,200	35,100	34,600	33,200
Power	213,300	287,300	261,400	252,900	235,300
Other					
Ethylene Product (lb moles/hr)	302.4	309.4	309.4	309.4	309.4
Ethylene Saving ($/yr)	-	48,000	48,000	48,000	48,000
Total Compressor HP	5,090	6,903	6,301	6,090	5,680
Power (kWh/Ton)	897	1190	1086	1050	979
TOTAL COST/YR	1,210,000	1,469,000	1,395,000	1,363,000	1,309,000

can be recovered as a significant proportion of it is lost in column irreversibilities.

iii) Another significant factor is the extra cost of around \$100,000/year required by the low pressure process for heat exchangers. This is primarily attributable to the close temperature approaches and the expensive low temperature construction materials.

There is one advantage of low pressure operation that should be noted. This is the increased ethylene recovery, estimated to be worth around \$48,000/year, at 3¢/lb. The major ethylene loss in high pressure operation is in the demethanizer overhead tail gas. The quantity of this loss is determined mainly by the available refrigerant temperature to the partial condenser. This in turn is set by the lowest practical ethylene refrigerant evaporation temperature to around -165°F. A somewhat lower temperature is attainable by cooling with expanded tail gas as described by King et al.[11], or by addition of a methane refrigerant circuit, neither of which were considered in the present study. In the low pressure process the demethanizer loss is much reduced as not only is the methane/ethylene separation easier at the lower demethanizer pressure but a suitably colder methane overhead refrigeration system is available. This lower demethanizer loss is only slightly offset by a small extra loss introduced by pre-separation of the tail gas from the demethanizer feed.

CHAPTER 8

PHYSICAL PROPERTIES CALCULATION

The present program system has depended on the CHESS[4] package for physical property estimation. It will be remembered that the package consists of a properties calculation routine which estimates single phase (liquid or vapor) properties and an isothermal/adiabatic flash routine which handles two-phase mixtures. As part of a process design program system such a package must meet three major requirements. It must (i) be completely reliable and (ii) be sufficiently accurate over a wide range of stream conditions without (iii) consuming an excessive proportion of the overall computing time. The present application studies have covered temperature and pressure ranges of approximately -250^{o}F to +200^{o}F and 1 at to 40 at respectively for components from hydrogen to butane. This is to the author's knowledge the first really extensive test of the CHESS package. Over this wide span it has in general proved most satisfactory. With one exception, described later, the package has provided property values accurate enough to permit the sophistication of equipment calculation necessary for realistic process design. It is estimated that around 30% of computation time is used in property estimation. There have however been several areas of difficulty necessitating modification to the original package and these are detailed below.

i) The original package used an iterative technique to establish the (compressibility) root of the cubic equation resulting from the Redlich-Kwong equation of state. It appeared that the wrong root was found under some conditions so that this iterative procedure was satisfactorily replaced by an analytical root finding routine.

ii) Convergence procedures for dew and in particular bubble points were not satisfactory especially as mixture critical points were approached. This was found to be due mainly to poor initial estimates of the composition of the other phase in equilibrium with the stream in question and produced an incorrect bound for the reguli-falsi iteration. The problem was solved by introducing an automatic re-start procedure to be used when problems were detected during iteration. Speed of covergence was further improved by addition of a simple procedure for approximate estimation of bubble and dew points from a regression equation involving mixture pressure and average molecular weight.

iii) Covergence routines for the isothermal and adiabatic flash routines also required modification. Problems stemmed largely from poor starting estimates. With the availability of bubble and dew point temperatures in the stream list good starting estimates were readily obtained by interpolation between these single-phase bounds and the resulting bounded reguli-falsi iteration scheme worked well.

iv) In only one instance did unsatisfactory property estimation result from deficiencies in the correlations rather than in the convergence procedures. This occurred in estimation of bubble/dew points for pure methane in the low temperature refrigeration circuit. Significantly low dew point estimates were obtained at pressures above around 100 psia and above 300 psia meaningful values could not be obtained. The problem, which appears to result from poor liquid activity coefficient estimation, is that K values for the pure component reach a minimum with respect to temperature which is greater than unity.

v) Only one addition was made to the package. This was to allow estimation of bubble and dew point pressures for a given temperature, required in vapor recompression and refrigeration unit calculations. It was implemented simply by changing the iteration variable from temperature to pressure in the bubble and dew point iteration procedures.

CHAPTER 9

CONCLUSIONS AND RECOMMENDATIONS

9.1 Conclusions

i) The branch and bound optimization technique has been found most effective for the present type of discrete, sequential processing problem. Especially with the automatic bounding problem selection the necessary logic has been readily incorporated into the computerized synthesis system. As was shown in the final process application, the branch and bound concept may be employed rather more freely in synthesis problems than was originally demonstrated by Lee et al.[10].

ii) The incorporation of heuristic decision making into the system has been shown to be very valuable both in reducing problem sizes and in permitting design experience to be embodied in the logical structure of the design system. The effectiveness of the branch and bound/heuristic combination has been demonstrated in the evolution of very realistic process designs.

iii) Price-based process decomposition has been shown to be of particular value in discrete, combinatorial synthesis problems. Not only is it useful in limiting problem sizes by creation of independent sub-processes and subsequent determination of optimal interconnections between them; it can also be used to prevent unwanted and/or unnecessary stream interactions between sub-processes. Integration of existing processes should also be possible by this method. Implementation of the method by use of prices based on real process costs and modified by the discount (δ) parameter has been demonstrated by a rather simple decomposition example. However the simplicity and flexibility of the technique combined with its potential for reduction in dimensionality and maintenance of overall process feasibility should recommend its application to rather more complex situations.

iv) The synthesis program system created (OPENS) is modularly oriented and with its stream number/equipment number based data structure it should be readily understood and easily used by the design engineer. However it must be emphasized that unlike some comparable simulation systems the executive program cannot be treated as a "black box" but rather as a "hands on" system which demands both decision making and programming input from the user. This is largely attributable to the equipment-dependent decisions which must be made within the system and the variability of the heuristic set which must be programmed into system routines. Due to the very nature of realistic synthesis, certainly at this early stage of its development, the same generality as has been achieved in simulation cannot be expected.

v) One feature of this study has been to elucidate further the possible range of creative capability and levels of decision making between simulation and synthesis. The current upper level is probably represented by the work of Siirola and Rudd[7] on evolution of basic processing schemes. The present OPENS system has dealt with synthesis of process equipment networks within such a pre-defined processing scheme. A still lower level is represented by the refrigeration unit routine which essentially generates equipment networks according to a pre-determined pattern. It has some limited decision making capability. At the lowest level are the familiar simulation systems which evaluate completely pre-determined equipment configurations. It is important to be aware of this range of approach if only to avoid the need for unnecessary creative sophistication in solution of any particular problem or class of problems.

9.2 Recommendations

9.2.1 Improvements to Present System

There are a number of possible areas of improvement to the present OPENS system which may improve its efficiency and extend its usefulness. These areas are detailed below.

i) As has been described earlier the present system stores a minimum of process information thus minimizing core storage but necessitating some increase in computation time for regenerating certain information. A particular example is the stream "history" information required for feasibility checking. It is suggested that if such information were retained in some form and perhaps more use were made of it than at present the somewhat involved branch and bound logic may be able to be simplified and its efficiency improved.

ii) The whole area of heuristics warrants a considerable amount of further study particularly if increasingly realistic process designs are to be developed. One particularly relevant area for improvement is that concerning the identification of dynamic problems particularly in process start-up and control. These problems are particularly difficult to forsee during the synthesis procedure but are very important to the successful operation of any synthesized process. In the absence of any systematic identification method a form of heuristic decision making applied to stream matching, as used in this study, is probably the best hope at present.

One possible direction for improvement, not just relevant to dynamic problems, is the introduction of scoring or penalty functions for stream matching in a manner similar to their present use in evaluating moves in game playing[35]. Points could be awarded for both individual stream and stream match characteristics with a certain accumulation of points necessary for a match to be acceptable. For example reflux streams could be penalized in comparison with product streams as regards the possible flexibility in their usage. Matches involving too great a degree of process feedback could be penalized from a start-up viewpoint. Too great an entropy increase in stream exchanger matching should also be penalized, etc. It may also be possible to introduce some degree of automated learning with such a numerical scale for evaluating stream matches.

iii) The use of the exergy concept in stream pricing may be able to be extended in certain situations where an accurate, comprehensive physical cost basis is not as readily available as in the present study. Most processes will have some associated service facilities and although these may provide some basis points for costing, the exergy function may still be useful for extrapolation and/or interpolation.

iv) A systematic price (δ) adjustment algorithm was not really required for the applications described but should be considered for future, more complex situations. The piecewise-constant nature of the process optimum cost vs δ function (refer to Figure 13) makes any guarantee of optimality difficult, since the use of any finite step in δ makes it possible that solutions may be missed. However for practical purposes the following algorithm is suggested for the general multi-dimensional (δ) case. It can be argued intuitively and from the present computational results that the cost function (of δ) is unimodal. Thus it is suggested that the practical δ space (say from -5 to +10 times ΔT_{min}) first be explored by grid search using a comparatively coarse grid. The region of the indicated optimum could then be isolated and explored using a progressively smaller grid, continuing until the user is satisfied with the accuracy obtained.

v) The potential for interactive operation should also be explored as the opportunity to complement the computer's computational speed with the human designer's decision making ability should considerably increase the power and flexibility of the system. It should be noted that the system has been run in four separate batch sections in this study and this has permitted a certain valuable degree of user interaction.

9.2.2 Extension of Applications of Present System

The present OPENS system is in principle capable of handling any discrete, sequential processing problem which can be defined in terms of temperature and pressure specifications for known streams. In an attempt to explore the wider potential of the system several process areas of application are examined below.

i) Gas Separation Processes

The most obvious applications are to other similar low temperature gas separation processes whether they are other ethylene plants, natural gas plants, etc. The system should be capable of handling such processes with little or no modification.

There is however one potential difficulty which should be examined. The system presently handles sequential processing problems for which overall mass and heat balances can be made prior to the application of the synthesis procedure, i.e., these balances must be independent of the processing method or sequence. There are two process sections in gas separation processes where this condition may not apply.

Firstly consider a stream (usually a feed) being cooled in an exchanger train with condensate being removed between cooling stages. Then, although this situation may be regarded as a sequential process, the mass and heat balances depend on the sequence and levels of cooling as these determine the changing flow profile through the train. This occurs for example in demethanizer feed cooling trains as described by King et al.[11], and would strictly require an iterative computation in which the effects of each cooling sequence on the subsequent balances would need to be determined. Note that it would also violate the current system requirement of constant stream composition, although this could be changed moderately easily. The second situation occurs in vapor recompression condensation of a column overhead stream. The stream is compressed in order to be able to condense it against some other stream, usually the same column bottom stream. However after condensation at this increased pressure a portion of this stream will be fed back to the column as liquid reflux. The decrease in pressure on column entry will result in flash vaporization of a certain quantity of liquid and this additional vapor flow is added to the existing column overhead stream. Thus a recycle situation develops which strictly necessitates iterative calculations. This difficulty has been sidestepped in the present examples by setting a fixed fraction flash-off (10%) for the returning liquid condensate.

ii) Air Separation Processes

The other major low temperature gas separation area is that of air separation[36]. Although these processes involve many of the same energy recovery considerations as in ethylene plants, closer examination shows considerable difficulties in potential application of the OPENS system. The process temperatures are rather lower than in ethylene plants so that very high energy efficiencies are required. In order to achieve this efficiency modern plants tend to be extremely complex with considerable equipment specialization, e.g., non-conventional columns, multi-fluid heat exchangers, etc. Also, partly as a result of these factors, it may be difficult to specify stream temperature requirements accurately in advance.

Attainable temperatures may depend on the other process streams available since no service refrigerants are used. Thus it may be difficult to define the problem adequately without prior knowledge of the actual final configuration. The considerations involved in process start-up and control may also play a large part in determining process configuration.

In view of these factors it does not appear that air separation processes presently present a very worthwhile area for system application. The evolutionary approach of King et al.[11] is probably better suited to this type of application.

iii) High Temperature Heat Recovery Processes

Another potential area of application is to high temperature heat recovery systems. Possible examples include the "front end" of ethylene plants, the reactor sections of ammonia plants and refinery steam systems. However there are a number of difficulties here as well.

There is again the problem of equipment specialization, as instead of using conventional countercurrent heat exchangers, heat may be generated and/or recovered in furnaces, boilers or even scrubbing towers.

A major consideration in these high temperature processes is that heat is frequently recovered by steam generation, i.e., by using evaporating water as a heat sink (such a provision would have to be made in the program for dealing with such systems) rather than by process/process exchange. This has considerable advantages from an operating flexibility viewpoint since steam can be distributed much more freely than can the streams that create it. However it effectively reduces the number of interacting energy levels to the point where the problem may not be large enough to warrant the application of branch and bound based synthesis. In fact, particularly in steam systems, the considerations involved in choosing the number and magnitude of the steam levels may be rather more important than those involved in determining system configuration[37]. Thus the problem may often involve a mixture of discrete and continuous decision making.

In view of the relative inflexibility in process configuration and the equipment specialization, the design of many such process networks may be handled better by the type of skeleton flowsheet generator approach as used in the refrigeration unit. Such a generator may serve as an objective function evaluator for the discrete and continuous design optimization calculations. The use of the OPENS system approach in high temperature processes should not however be ruled out.

One related area that is worth investigating is that of multistage evaporator networks. Batstone and Prince[6] have reported on the design of such systems using a repetitive simulation-based approach but it appears that the branch and bound synthesis technique may be able to lead directly to optimal process configurations.

iv) Summary

In view of the above examinations it would appear that the OPENS type of synthesis system does not have the breadth of usefulness that was hoped at the outset of the study. However these are some areas that should at least be worth investigating. Modifications to both executive and equipment routines may be necessary in many cases and the application of the system to any potential area is of course dependent on the provision of a suitable physical properties package.

9.2.3 Wider Extensions

There are two further areas which, although beyond the scope of the present study, may prove worthy of future investigation.

i) The present study has gone beyond previous energy exchange network studies in allowing for stream pressure changes. However pressure has been regarded only as a thermal energy level modifier and the broader implications of pressure energy recovery as such have not been explored. There are a number of process areas where such energy recovery is important, e.g., in natural gas processing. Pressure energy is usually recovered by turbo-expanders which may at the same time achieve required process stream cooling. The pricing of such streams embodying pressure and/or thermal energy may provide a further opportunity for use of the generalized exergy concept. This may require the provision of generalized entropy calculation within the physical properties package.

ii) It would be useful to examine the possibility of integrating the present energy exchange network synthesis system with the work of Thompson and King[12] on separation process synthesis. Their work encompasses many similar principles to those used in the present study. It includes a significant heuristic element and the optimum separation sequencing problem is essentially a combinatorial one. Thus such a combination of techniques would be a significant step towards complete process synthesis.

The whole field of systematic, automated process synthesis has only begun to be explored and a great deal of additional work is required in many areas before the real creative capability of the computer in this field can be gauged. It is particularly important to learn which tasks can best be accomplished by both the computer system and the human designer as it is by such a merging of capabilities that the most efficient synthesis methods will be evolved.

CONTRIBUTIONS

In the author's estimation this study has made the following contributions to engineering knowledge.

i) A flexible computer system for synthesis of optimal energy exchange systems has been developed. It is capable of handling both stream temperature and pressure demands. The development of this system has involved:

(a) An extension and greater understanding of the branch and bound combinatorial optimization strategy.

(b) An effective combination of branch and bound strategy with heuristic decision making.

(c) An extension of price-oriented, Lagrange Multiplier decomposition techniques to large discrete, combinatorial process design problems.

ii) The usefulness of the "modular approach" in process design has been significantly extended by incorporating within it a creative capacity for automated synthesis.

iii) To the author's knowledge, this study has produced the first automated, optimal synthesis of a complex real chemical process and has thus demonstrated the practical capabilities of the approach.

The following publications have been produced during this study:

1. "Synthesis of Optimal Energy Recovery Networks using Discrete Methods", accepted for publication by Can. J. Ch.E.

2. "Synthesis of Optimal Energy Recovery Networks using Discrete Methods", to be presented at the 71st National AIChE meeting, Dallas (Feb., 1972).

NOMENCLATURE

a	Temperature difference
D	Design
f	Sub process cost function
F	Overall process cost function
H	Stream enthalpy
m	Discrete decision variable subset
M	Discrete decision variable set
N	Number of processing path combinations (networks)
O	Objective function
p	Stream processing path
pr	Stream price function
P	Stream transfer price
S	Stream entropy
T	Stream temperature
X	Flow of transferred stream
δ	Stream pricing "discount" parameter
ε	Stream exergy

REFERENCES

1. Crowe, C. M., Hamielec, A. E., Hoffman, T. W., Johnson, A. I., Shannon, P. T. and Woods, D. R., "Chemical Plant Simulation", Prentice Hall (1971).

2. Westerberg, A. W., and Edie, F. C., "Computer Aided Design", Chem.Eng.J., 2, No. 1, 9 (1971).

3. Johnson, A. I., and Associates, "GEMCS Manual and Application Studies", Dept. of Chem. Eng., McMaster University, Hamilton, Ontario (1970).

4. Motard, R. L., Lee, H. M., Barkely, R. W., and Ingels, D. M., "CHESS, Chemical Engineering Simulation System, System Guide", Tech. Publ. Co.,4375 Harvest Lane, Houston, Texas (1968).

5. Harris, R. E., "FLOWTRAN: An Approach to Computer Aided Process Analysis and Design Tool", Computerized Engineering Dept., Monsanto Company (1971).

6. Batstone, D. B. and Prince, R. G. H., "Computer-Aided Design and its Application to Planning Steam Systems in Sugar Factories", Presented to CHEMECA '70 Conference, Melbourne and Sydney, (August, 1970).

7. Siirola, J. J. and Rudd, D. F., "Computer-Aided Synthesis of Chemical Process Designs", I&EC Fundamentals, 10, No. 3, 353 (1971).

8. Kessler, M. G. and Parker, R. O., "Optimal Networks of Heat Exchange", CEP Symp. Series, 65 No. 92, 111 (1969).

9. Masso, A. H. and Rudd, D. F., "The Synthesis of System Designs, II. Heuristic Structuring", A.I.Ch.E.J., 15, No. 1, 11 (1969).

10. Lee, K. F., Masso, A. H. and Rudd, D. F., "Branch and Bound Synthesis of Integrated Process Designs", I&EC Fundamentals, 9, No. 1,48 (1970).

11. King, C. J., Gantz, D. W. and Barnés, F. J., "Systematic Evolutionary Process Synthesis", Presented to Am.Chem. Soc. Annual Mtg.,Los Angeles (March, 1971).

12. Thompson, R. W. and King, C. J., "Systematic Synthesis of Separation Schemes", to be presented at A.I.Ch.E. 71st National Mtg., Dallas, Texas (February, 1972).

13. Lasdon, L. S., "Optimization Theory for Large Systems", MacMillan (1970).

14. Brosilow, C. and Nunez, E., "Multi-Level Optimization Applied to a Catalytic Cracking Plant", Can. J. Chem. Eng., 46, 205 (June, 1968).

15. Gembicki, S. A., "Modular Optimization of Chemical Processes", Ph.D. Thesis, Thayer School of Engineering, Dartmouth College, Hanover, New Hampshire (1969).

16. Everett, H.,"Generalized Lagrange Multiplier Method for Solving Problems of Optimal Allocation of Resources", Operations Research, 11, 399 (1963).

17. Ahlberg, J. H. et al., "The Theory of Splines and Their Applications", Academic Press, New York (1967).

18. Tribus, M. and Evans, R. B., "Thermo-Economics of Saline Water Conversion", I&EC Proc. Des. and Dev., 4, No. 2, 195 (1965).

19. Chao, K. C., and Seader, J. D., "A General Correlation of Vapor-Liquid Equilibria in Hydrocarbon Mixtures", A.I.Ch.E.J., 7, No. 4, 598 (1961).

20. Grayson, H. G. and Streed, C. W., Paper 20-PD7, Sixth World Petroleum Conference, Frankfurt (June, 1963).

21. Yen, L. C. and Woods, S. S., "A Generalized Equation for Computer Calculation of Liquid Densities", A.I.Ch.E.J., 12, No. 1, 95 (1966).

22. Hengstebeck, R. J., "An Improved Shortcut for Calculating Difficult Multicomponent Distillations", Chem. Eng., 76, No. 1, 115 (1969).

23. King, C. C., "The Low Temperature Separation of Hydrocarbons", T.I.Ch.E., 36, 162 (1958).

24. Bauman, C. H., "Fundamentals of Cost Engineering in the Chemical Industry", Reinholt Publ. Co. (1964).

25. Peters, M. S. and Timmerhaus, K. D., "Plant Design and Economics for Chemical Engineers", 2nd Ed., McGraw-Hill (1968).

26. Hand, W. E., "From Flowsheet to Cost Estimate", Petr. Ref., 37, No. 9, 331 (1958).

27. Clancy, G. M. and Townsend, R. W., "Ethylene Plant Fractionation", CEP, 67, No. 2, 41 (1971).

28. Aalund, L., "Houston's Big Ethylene Plant Marks a New Era in Olefins", Oil and Gas J., 158, (Nov. 20, 1967).

29. Charlesworth, P. L., "The Production of Light Olefins", Brit. Chem. Eng., 10, No. 12, 834 (1965).

30. Baldus, H. and Linde, G., "Tieftemperaturzerlegung Kohlenwasserstoffreicher Gase zur Gewinnung Von Äthylen und Propylen", Kaltetechnik, 15, No. 6, 158 (1963).

31. Brooks, K. W., "Low-Temperature Separation for Ethylene Recovery", Oil and Gas J., 70 (Dec. 21, 1964).

32. Ruhemann, M. and Charlesworth, P. L., "The Thermodynamic Efficiency of Gas Separation Plants", Brit. Chem. Eng., 11, No. 8, 839 (1966).

33. Haselden, G. G., "An Approach to Minimum Power Consumption in Low Temperature Gas Separation", T.I.Ch.E., 36, 162 (1958).

34. Perry, J. H. (Ed.), "Chemical Engineers' Handbook", 4th Ed., p. 18.25 (1963).

35. Dale, E. and Michie, D. (Eds.), 'Machine Intelligence 2", Am. Elsevier Publ. Co. (1968).

36. Latimer, R. E., "Distillation of Air", CEP, 63, No. 2, 35 (1967).

37. Arnstein, R. and O'Connell, L., "What's the Optimum Heat Cycle for Process Utilities", Hydr. Proc., 47 No. 6, 88 (1968).

APPENDICES

APPENDIX I

DERIVATIONS

I.1 Stream Energy Value Integration

From equation 14, section 3.3, the value of any stream between temperature limits T_1 and T_2 is given by the integration

$$P = \int_{T_1}^{T_2} pr(\theta \pm \delta) \left(\frac{dH}{d\theta}\right) d\theta \qquad (I.1)$$

The function pr is defined by a known cubic spline in T and the value and sign of the δ parameter are known. H is the stream enthalpy.

The integration range is first divided into its various phase regions (liquid, two-phase and vapor). Over each region it may be assumed that the differential dH/dT, which is actually the stream specific heat, is approximately constant, i.e. H is a linear function of T. Then for each phase segment the integration, between limits A and B, becomes

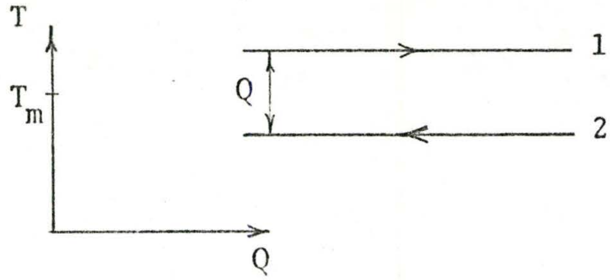
$$P_{AB} = \frac{\Delta H_{AB}}{\Delta T_{AB}} \cdot \int_{T_A}^{T_B} pr(\theta \pm \delta) d\theta \qquad (I.2)$$

This function can then be integrated numerically using Simpson's Rule, as follows:

$$P_{AB} = \frac{1}{6} \Delta H_{AB} \left[pr(T_A \pm \delta) + 4pr\left(\frac{T_A+T_B}{2} \pm \delta\right) + pr(T_B \pm \delta)\right] \qquad (I.3)$$

The contributions of each phase segment can then be assumed to give the total stream value.

I.2 Approximate Expression for Heat Exchanger Entropy Increase



Consider the heat transfer between an infinite source and sink, 1 and 2, as shown above. The mean temperature is T_M and the temperature driving force for exchange is a. The entropy change for the process for transfer ΔQ is given by:

$$\Delta S = -\frac{\Delta Q}{T_1} + \frac{\Delta Q}{T_2}$$

$$= -\frac{\Delta Q}{T_M + \frac{a}{2}} + \frac{\Delta Q}{T_M - \frac{a}{2}} \qquad (I.4)$$

Expressed on a per unit heat transfer basis the expression becomes:

$$\frac{\Delta S}{\Delta Q} = \frac{-1}{T_M + \frac{a}{2}} + \frac{1}{T_M - \frac{a}{2}} \qquad (I.5)$$

which simplifies to

$$\frac{\Delta S}{\Delta Q} = \frac{a}{T_M^2 - \frac{a^2}{4}} \qquad (I.6)$$

and since the temperature difference, a, will be much smaller than the absolute temperature T_M, then

$$\frac{\Delta S}{\Delta Q} \doteqdot \frac{a}{{T_M}^2} \tag{I.7}$$

In the more general case where temperatures are not constant throughout the exchange process the expression, I.7, remains a useful approximation.

APPENDIX II

PROGRAM SYSTEM

II.1 Program Descriptions and Listings

The OPENS program system is presently run in four sections, both for greater operating flexibility and reduced central memory requirement. Each of these sections has a main program, primarily for data input purposes, which then calls a series of subroutines to carry out the appropriate system functions. The program make-up of the four sections is detailed below.

A Task Identification (Column Calculation)	MAINC
	COLSYS, STMOVC
B Stream Processing Path Generation	MAINS
	SMATCH, STMOVS, SHIST
C Selection of Optimal Network Configuration	MAINB
	ENERGY
	BRBND, (STMOVS, SHIST)
D Refrigeration Unit	MAINR
	RUNIT, STMOVR

There are two further groups of subroutines, those for physical properties calculation and handling, and those for equipment calculation. They are listed below.

E Physical Properties	PROPL, KHZT, FLASH, ZERO, ZEROI

F Equipment Routines

(FLASH)
DIST
HXER
COMP
SPLIT
SPLINE, SVALUE, INTER

Note that the flash routine, FLASH, serves both as a two-phase properties estimation routine and as an equipment routine for adiabatic expansion and mixing.

The final three routines above (SPLINE, SVALUE and INTER) are all involved with the spline-based stream pricing scheme.

Brief program descriptions and full listings follow in the order detailed above.

The programs are set up as for the high pressure process case study and on the few occasions where statements are specific to that particular study this is indicated by "HP" in the card identification field (columns 73, 74).

A TASK IDENTIFICATION

MAINC reads the input data for this section. It also performs the initialization functions of pre-zeroing appropriate matrices and computing input stream bubble and dew point temperatures and enthalpies.

COLSYS is the simulation-type executive for column system calculation. It computes the specified column configuration (coded in process matrix form) in sequence, performing an overall mass and heat balance and computing flows and properties for intermediate and output streams. Streams with unsatisfied pressure, phase and temperature demands are identified by comparing their properties with supplied specifications. Any phase specifications are converted into corresponding temperature specifications. Streams are classified as either "hot" or "cold" (requiring cooling or heating) for later stream matching purposes. In addition to dealing with supplied stream specifications, two special demands are automatically created for each column, those for overhead and bottom reflux generation. The program finally produces a punched deck of stream specifications and properties which serves as input to the following stream processing path generation section.

STMOVC is the stream handling utility routine which moves stream properties information between the stream matrices and working vectors.

```
      PROGRAM MAINC(INPUT=1001,OUTPUT=1001,PUNCH=1001,TAPE5=INPUT,TAPE6=
     1OUTPUT,TAPE7=PUNCH)
C
C      *** COMMON DECK ***
C
C      DUMMY ARRAYS ARE FOR NAMELIST INPUT
C
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/KPM/NIS,NKPM,KPM1(6),KPM2(6),KPM3(6),KPM4(6),KPM5(6),
     1 KPM6(6),KPM7(6),KPM8(6),KPM9(6),KPM10(6)
      COMMON/SMP/SMPA1(8),SMPA2(8),SPMA3(8),SMPA4(8),SMPA5(8),
     1 SMPA6(8),SMPA7(8),SMPA8(8),SMPA9(8),SMPA10(8),SMPA11(8),
     2 SMPA12(8),SMPA13(8),SMPA14(8),SMPA15(8),SMPA16(8),SMPA17(8),
     3 SMPA18(8),SMPA19(8),SMPA20(8),SMPA21(8),SMPA22(8),SMPA23(8),
     4 SMPA24(8),SMPA25(8),
     4 SMPB1(15),SMPB2(15),SMPB3(15),SMPB4(15),SMPB5(15),SMPB6(15),
     5 SMPB7(15),SMPB8(15),SMPB9(15),SMPB10(15),SMPB11(15),SMPB12(15),
     6 SMPB13(15),SMPB14(15),SMPB15(15),SMPB16(15),SMPB17(15),
     7 SMPB18(15),SMPB19(15),SMPB20(15),SMPB21(15),SMPB22(15),
     8 SMPB23(15),SMPB24(15),SMPB25(15)
      COMMON/EMI/EMI1(15),EMI2(15),EMI3(15),EMI4(15),EMI5(15),EMI6(15),
     1 EMI7(15),EMI8(15),EMI9(15),EMI10(15)
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
      COMMON/PROP/COMPNT(8),APC(8),ATC(8),AVC(8),AMW(8),AOMEG(8),
     1 ADEL(8),AVW(8),APH(8),BET(8),GAM(8),DTA(8),BASEA(8),BASEB(8),
     2 ZCD(8),ALD(8)
      COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR
C      ***
C
      INTEGER COMPNT
      DIMENSION KPM(6,10),SMPA(8,25),SMPB(15,25),EMI(15,10)
      DIMENSION TITLE(8),DUM(8),PROP(8,15)
      EQUIVALENCE (KPM,KPM1),(SMPA,SMPA1),(SMPB,SMPB1),(EMI,EMI1)
      EQUIVALENCE (PROP,APC)
C
      NAMELIST/KPMLST/NIS,NKPM,KPM1,KPM2,KPM3,KPM4,KPM5,KPM6,KPM7,KPM8,
     1 KPM9,KPM10
      NAMELIST/SMPLST/SMPA1,SMPA2,SMPA3,SMPA4,SMPA5,SMPA6,SMPA7,
     1 SMPA8,SMPA9,SMPA10,SMPA11,SMPA12,SMPA13,SMPA14,SMPA15,SMPA16,
     2 SMPA17,SMPA18,SMPA19,SMPA20,SMPA21,SMPA22,SMPA23,SMPA24,SMPA25,
     3 SMPB1,SMPB2,SMPB3,SMPB4,SMPB5,SMPB6,SMPB7,SMPB8,SMPB9,SMPB10,
     3 SMPB11,SMPB12,SMPB13,SMPB14,SMPB15,SMPB16,SMPB17,SMPB18,SMPB19,
     4 SMPB20,SMPB21,SMPB22,SMPB23,SMPB24,SMPB25
      NAMELIST/EMILST/EMI1,EMI2,EMI3,EMI4,EMI5,EMI6,EMI7,EMI8,EMI9,EMI10
      NAMELIST/PARLST/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR
      NAMELIST/COMP/NOCOMP,COMPNT
C
C      PRE-ZERO ARRAYS
      CALL ZEROI(KPM,60)
      CALL ZERO(SMPA,725)
C
C      READ COMPONENT INFORMATION
      READ(5,COMP)
C      READ COMPONENT PHYSICAL CONSTANTS
      DO 10 I=1,NOCOMP
   10 READ(5,11)(PROP(I,K),K=1,15)
C
C      READ TITLE
      READ(5,100)TITLE
      IF(EOF,5)1,2
    1 CALL EXIT
    2 WRITE(6,101)TITLE
C      READ GENERAL SYSTEM PARAMETERS
      READ(5,PARLST)
C      READ PROCESS MATRIX
      READ(5,KPMLST)
C      READ STREAM MATRICES (A+B SECTIONS)
      READ(5,SMPLST)
C      READ EQUIPMENT MATRIX
      READ(5,EMILST)
C
```

```
C     CALCULATE INPUT STREAM BUBBLE , DEW POINTS + ENTHALPIES
      DO 20 J=1,NIS
      IF(SMPB(7,J).EQ.0.) GO TO 20
      CALL MVFSM(1,J,0,0)
      CALL BUBTP(1,SMPB(1,J),DUM)
      CALL DEWTP(1,SMPB(2,J),DUM)
      NIN=NOUT=1
      CALL ISOF(0.)
      SMPB(5,J)=HOUT(1)
      SMPB(6,J)=VFOUT(1)
   20 CONTINUE
      CALL COLSYS
C
   11 FORMAT(3(/5E14.5))
  100 FORMAT(8A10)
  101 FORMAT(8A10//)
      END
```

```
      SUBROUTINE COLSYS
C
C     COMPUTES COLUMN SYSTEM , IDENTIFIES HEAT TRANSFER + EXPANSION/
C     COMPRESSION TASKS AND MAKES ENTRIES IN STREAM SPEC VECTORS
C
C
C     KPM  PROCESS MATRIX CODING ..
C     FOR EACH (EQUIPMENT) VECTOR OF KPM -
C        1.      EQUIPMENT NUMBER
C        2.      EQUIPMENT TYPE NUMBER
C        3.-6. INPUT , THEN OUTPUT STREAM NUMBERS (+ INPUT , - OUTPUT)
C
C     SMPA  STREAM CONTROL VECTOR CODING -
C        1. STREAM NO
C        2. STREAM TYPE - 1. FEED , 0. INTERMEDIATE , -1. PRODUCT
C        3. FURTHER PROCESSING FLAG - 1. FOR FURTHER PROCESSING
C        4. PRESSURE SPEC
C        5. PHASE SPEC  (1.+VAPOR FRACTION)
C        6. TEMPERATURE SPEC
C
C*****  COMMON DECK
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/KPM/NIS,NKPM,KPM(6,10)
      COMMON/SMP/SMPA(8,25),SMPB(15,25)
      COMMON/EMI/EMI(15,10)
      COMMON/EQUIP/EQUIP(15)
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
      COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR
C
      DIMENSION DMCHA(8),DUM(8),NSCH(10,2),FSAV(4,2),NNPR(2)
C     INITIALIZE MATRIX COUNTERS
      NNPR(2)=NNPR(1)=0
      NSM=NIS
C*
      DO 60 NE=1,NKPM
C     FIND NO OF INPUTS (NI)
      DO 4  J=3,6
    4 IF(KPM(J,NE).LE.0) GO TO 6
    6 NI=J-3
      JF=3
      JL=NI+2
      JIO=1
C
C     SCAN EQUIP STREAMS FOR DEMANDS
    8 DO 12 JS=JF,JL
      NS=JIO*KPM(JS,NE)
      NF=JS-2
      IF(SMPA(3,NS).EQ.0.) GO TO 12
      JTYPE=SMPA(2,NS)
C     FOR INTERMEDIATE STREAMS (JTYPE=0) PROCESS ONCE ONLY - AS EQUIP ..
C     INPUTS - I.E. DONT PROCESS AS EQUIP OUTPUTS
      IF(JTYPE.EQ.0.AND.JIO.EQ.-1) GO TO 12
      PRES=SMPA(4,NS)
      PHAS=SMPA(5,NS)
      TEMP=SMPA(6,NS)
C
C     SATISFY DEMANDS - MOVE STREAM INTO SIN(1) FOR PROCESSING
      CALL MVFSM(1,NS,0,0)
      ASSIGN 12 TO LOC
      GO TO 100
   12 CONTINUE
      IF(JIO.EQ.-1) GO TO 35
C
C     LOAD INPUTS FROM SMPB INTO SIN
      DO 20 JS=JF,JL
      NF=JS-2
      NS=KPM(JS,NE)
      CALL MVFSM(NF,NS,0,0)
      IF(SMPA(3,NS).EQ.0.) GO TO 20
C     SET POST-PROCESSED STREAM PROPS
      TIN(NF)=FSAV(1,NF)
      PIN(NF)=FSAV(2,NF)
```

```
      HIN(NF)=FSAV(3,NF)
      VFIN(NF)=FSAV(4,NF)
   20 CONTINUE
C
C     WRITE OUT INPUT
      WRITE(6,900)NE
      WRITE(6,901)BPIN(1),DPIN(1),TIN(1),PIN(1),HIN(1),VFIN(1),TMIN(1),
     1(XIN(I,1),I=1,NOCOMP)
C
C     COMPUTE EQUIP (COLUMN)
C*
      CALL DIST
C*
      PCOND=EQUIP(4)
      TCOND=EQUIP(5)
C
C     WRITE OUT COLUMN EQUIP VECTOR + OUTPUTS 1-4
      EQUIP(1)=NE
      WRITE(6,902)(EQUIP(I),I=1,15)
      DO 903 J=1,4
  903 WRITE(6,901)BPOUT(J),DPOUT(J),TOUT(J),POUT(J),HOUT(J),VFOUT(J),
     1TMOUT(J),(XOUT(I,J),I=1,NOCOMP)
C
C     LOAD OUTPUT STREAMS INTO SMPB , AS SPECIFIED IN KPM
      JF=NI+3
      DO 30 JS=JF,6
      NS=-KPM(JS,NE)
      IF(NS.EQ.0) GO TO 31
      NO=JS+1-JF
   30 CALL MVTSM(-NO,NS,0,0)
   31 JL=JF+NO-1
C
C     SCAN OUTPUTS FOR DEMANDS
C     FOR COLUMNS THERE ARE TWO SPECIAL OUTPUTS -
C         SOUT(3) TO BE CONDENSED  (OR P.C.)
C         SOUT(4) TO BE REBOILED
C
C     SCAN NORMAL OUTPUTS
      JIO=-1
      GO TO 8
C
C     SATISFY DEMANDS FOR SPECIAL OUTPUTS
   35 JTYPE=0
      DO 45 J=3,4
      TEMP=PRES=PHAS=0.
C     MOVE INTO SIN(1) FOR PROCESSING
      CALL MVSOSI(1,J,0,0)
      IF(J.EQ.4) GO TO 38
C     OVERHEADS VAPOR FLOW TO CONDENSER (TOTAL OR PARTIAL)
      IF(PCOND.LE.0) PHAS=1.
      IF(PCOND.EQ.1) TEMP=TCOND
      GO TO 40
C     BOTTOMS LIQUID FLOW TO REBOILER
   38 PHAS=2.
C     STORE VALUES IN SMP ARRAYS
   40 NSM=NSM+1
      SMPA(1,NSM)=NSM
      SMPA(4,NSM)=SMPA(2,NSM)=0.
      SMPA(6,NSM)=TEMP
      CALL MVTSM(1,NSM,0,0)
      NS=NSM
      ASSIGN 45 TO LOC
      GO TO 100
   45 CONTINUE
   60 CONTINUE
C
C
      WRITE(6,250)
      WRITE(6,200)NNPR
      WRITE(7,200)NNPR
      DO 300 ICH=1,2
      NNN=NNPR(ICH)
      WRITE(6,250)
      DO 202 J=1,NNN
```

```
      NS=NSCH(J,ICH)
      FJ=J
      WRITE(6,204)FJ,SMPA(2,NS),(SMPA(L,NS),L=4,6,2)
  202 WRITE(7,204)FJ,SMPA(2,NS),(SMPA(L,NS),L=4,6,2)
      WRITE(6,250)
      DO 206 J=1,NNN
      NS=NSCH(J,ICH)
      WRITE(6,250)
      WRITE(6,208)(SMPB(L,NS),L=1,15)
  206 WRITE(7,208)(SMPB(L,NS),L=1,15)
  300 CONTINUE
      RETURN
C
C
C**** DEMAND ROUTINE ****      OPERATES ON SIN(1)
C
C     IDENTIFY STREAMS AS HOT (ICH=1) OR COLD (ICH=2)
  100 CONTINUE
      ICH=2
      IF(TEMP.LE.0.) GO TO 102
      IF(TEMP.LT.TIN(1)) ICH=1
      GO TO 110
  102 PHASE=PHAS-1.
      IF(PHASE.LT.0.) GO TO 104
      IF(PHASE.LT.VFIN(1)) ICH=1
      GO TO 110
  104 IF(PRES.GT.PIN(1)) ICH=1
C
C     RECORD STREAM NUMBERS IN NSCH MATRIX + SMPA VECTORS
  110 NNS=NNPR(ICH)=NNPR(ICH)+1
      NSCH(NNS,ICH)=NS
      IVF=0
C
C**   PRESSURE
      IF(PRES.EQ.0.) PRES=PIN(1)
      IF(PRES.EQ.PIN(1)) SMPA(4,NS)=0.
C
C**   PHASE
C     TEMP SPEC OVERRIDES PHASE SPEC
      IF(PHASE.LT.0.) PHASE=VFIN(1)
      IF(TEMP.GT.0.) GO TO 120
      IF(PHASE.EQ.VFIN(1)) GO TO 120
C     SET DEFAULT VALUES FOR TEMP UNLESS -VE (TEMP FREE)
      IF(TEMP.LT.0.) GO TO 120
      IF(PHASE.EQ.0.) TEMP=BPIN(1)
      IF(PHASE.EQ.1.) TEMP=DPIN(1)
      SMPA(6,NS)=TEMP
C
C**   TEMP
  120 IF(TEMP.LE.0.) TEMP=TIN(1)
      IF(TEMP.EQ.TIN(1)) GO TO 130
C     SET FLAG FOR FINDING VF
      IVF=1
C
C*    SET POST-PROCESSED STREAM PROPS - FOR EQUIP INPUTS ONLY
  130 IF(JIO.NE.1) GO TO 150
      TIN(1)=TEMP
      PIN(1)=PRES
      IF(IVF.EQ.0) GO TO 140
      NIN=NOUT=1
      CALL ISOF(0.)
      CALL MVSOSI(1,1,0,0)
      GO TO 145
  140 VFIN(1)=PHASE
      CALL ENTH(1,HIN(1),DUM)
C     STORE POST-PROCESSED STREAM PROPS
  145 FSAV(1,NF)=TIN(1)
      FSAV(2,NF)=PIN(1)
      FSAV(3,NF)=HIN(1)
      FSAV(4,NF)=VFIN(1)
  150 GO TO LOC,(12,45)
C
  200 FORMAT(2I5)
  204 FORMAT(F6.0,12X,F6.0,19X,2F8.1)
```

```
208 FORMAT(3F8.1,5X,F8.1,F10.0,F7.3,F8.1/7F10.2)
250 FORMAT(//)
900 FORMAT(////* COL NO*,I5/)
901 FORMAT(3F8.1,5X,F8.1,F10.0,F7.3,F8.1,5X,7F6.1)
902 FORMAT(/* EQUIP VECTOR *,3F5.0,5X,3F8.2,2F6.0,2F7.2/F8.2,4F10.0/)
    END
```

```
      SUBROUTINE STMOVC(IWV,ISM,III,NX)
C
C     STREAM MOVING UTILITY ROUTINE ...(COLSYS VERSION)
C
C     IWV   - ELEMENT NUMBER IN SIN OR SOUT WORKING ARRAY
C             + SIN
C             - SOUT
C     ISM   - VECTOR NUMBER IN SM--
C     III     - 0     MOVE TO OR FROM SMPB
C               1-2   MOVE TO OR FROM SMCHB - (1-2)
C               3-5   MOVE TO OR FROM SMRB -  (1-3)
C     NX      - VECTOR NUMBER CONTAINING MOLE FRACTIONS  (III GT 0)
C
C     3 ENTRIES -
C        1 MVSOSI MOVES SOUT VECTOR ISM TO SIN VECTOR IWV (III=0)
C        2 MVFSM MOVES FROM SM-- TO SIN OR SOUT
C        3 MVTSM MOVES TO SM-- FROM SIN OR SOUT
C
C     *** COMMON DECK ***
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/SMP/SMPA(8,25),SMPB(15,25)
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
C     ***
      DIMENSION SMCHB(1,1,1),SMCHX(1,1,1),SMRB(1,1,1),SMRX(1,1,1)
      DIMENSION SIDUM(4,7),SODUM(4,7)
      EQUIVALENCE (BPIN,SIDUM),(BPOUT,SODUM)
C
C**
      ENTRY MVSOSI
      IENT=1
      GO TO 1
C**
      ENTRY MVFSM
      IENT=2
      GO TO 1
C**
      ENTRY MVTSM
      IENT=3
C
    1 JJJ=III+1
      GO TO (2,3,3,4,4)JJJ
    2 ITYPE=1
      GO TO 5
    3 ITYPE=2
      KKK=III
      GO TO 5
    4 ITYPE=3
      KKK=III-2
C
    5 GO TO (100,200,300)IENT
C
C**   MVSOSI
  100 DO 50 I=1,7
   50 SIDUM(IWV,I)=SODUM(ISM,I)
      DO 60 I=1,NOCOMP
   60 XIN(I,IWV)=XOUT(I,ISM)
      RETURN
C
C**   MVFSM
  200 DO 10 I=1,7
      GO TO (6,7,8)ITYPE
    6 AA=SMPB(I,ISM)
      GO TO 9
    7 AA=SMCHB(I,ISM,KKK)
      GO TO 9
    8 AA=SMRB(I,ISM,KKK)
    9 IF(IWV.GT.0) SIDUM(IWV,I)=AA
      IF(IWV.LT.0) SODUM(-IWV,I)=AA
   10 CONTINUE
C
      DO 20 I=1,NOCOMP
      GO TO (11,12,13)ITYPE
   11 AA=SMPB(I+7,ISM)
```

```
      GO TO 18
   12 AA=SMCHX(I,NX,KKK)*SMCHB(7,ISM,KKK)
      GO TO 18
   13 AA=SMRX(I,NX,KKK)*SMRB(7,ISM,KKK)
   18 IF(IWV.GT.0) XIN(I,IWV)=AA
      IF(IWV.LT.0) XOUT(I,-IWV)=AA
   20 CONTINUE
      RETURN
C
C**   MVTSM
  300 DO 30 I=1,7
      IF(IWV.GT.0) AA=SIDUM(IWV,I)
      IF(IWV.LT.0) AA=SODUM(-IWV,I)
      GO TO (21,22,23)ITYPE
   21 SMPB(I,ISM)=AA
      GO TO 30
   22 SMCHB(I,ISM,KKK)=AA
      GO TO 30
   23 SMRB(I,ISM,KKK)=AA
   30 CONTINUE
C
      IF(ITYPE.GT.1) RETURN
      DO 40 I=1,NOCOMP
      IF(IWV.GT.0) AA=XIN(I,IWV)
      IF(IWV.LT.0) AA=XOUT(I,-IWV)
   40 SMPB(I+7,ISM)=AA
      RETURN
      END
```

B STREAM PROCESSING PATH GENERATION

MAINS reads the input data for this section. This data consists mainly of the stream specification and properties information from the previous section.

SMATCH is the routine which generates the set of processing paths or equipment sequences for all primary streams. A description of its function has been given in section 4.2.2 and will not be repeated here. Instead a graphical algorithm is given in Figure II.1. There are several points of explanation which should be noted.

i) All pressure specifications (excluding vapor recompression, which is not actually a pressure specification) are met before proceeding with stream matching for satisfying temperature specifications.

ii) The major use of stream matching heuristics is in rejecting "type-infeasible" matches as indicated in Figure II.1. Some use is also made of heuristics in screening for vapor recompression matches. A C++++ card in the program listing indicates the use of heuristics.

iii) The test for match infeasibility due to multiple stream use is made using the SHIST stream history subroutine which is described below.

The major output from SMATCH is in the form of stream, stream processing path and equipment information which jointly define all possible process equipment network configurations. These data serve as input to the following section, C.

STMOVS is the version of the stream handling utility routine for this and the following section.

SHIST is the routine which generates "stream histories" (list of streams used in producing a given stream) which are used in identifying match infeasibility due to multiple stream use. These histories are generated from information in the stream, stream processing path and equipment arrays.

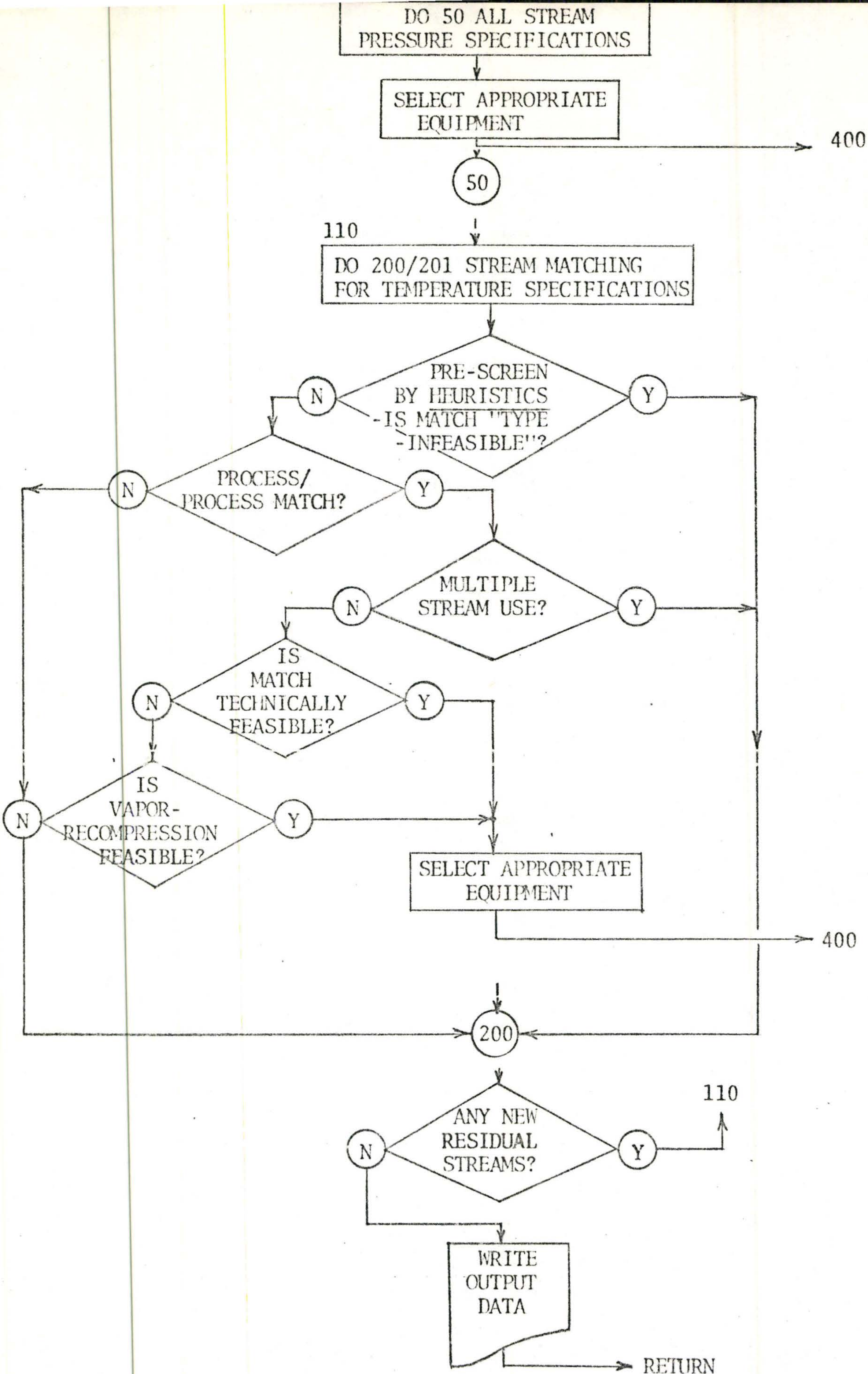


Figure II.1 SMATCH Algorithm (Continued on page 130).

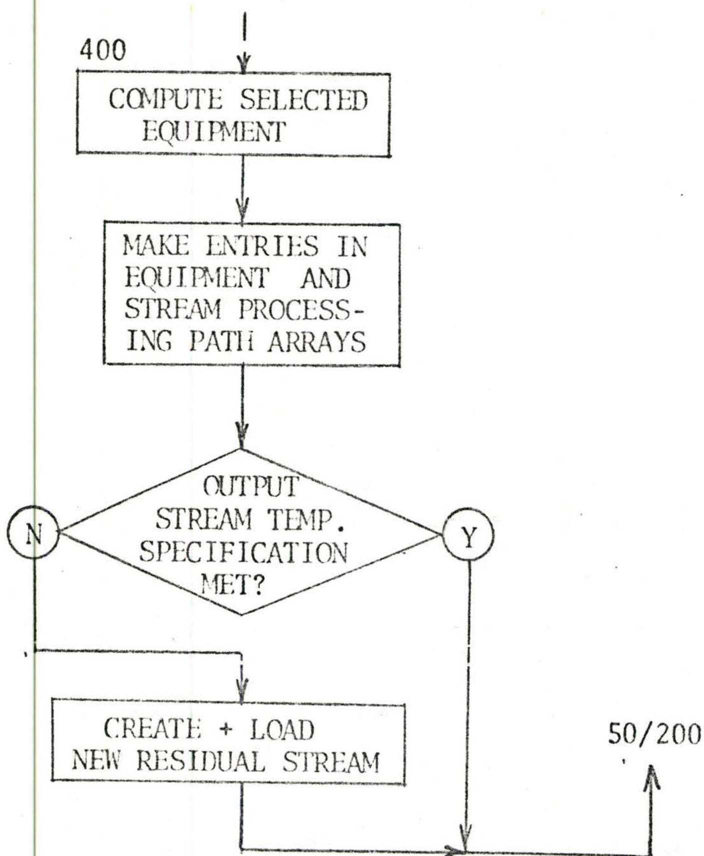


Figure II.1 SMATCH Algorithm (Continued).

```
      PROGRAM MAINS(INPUT=1001,OUTPUT=1001,DUMMY=1001,TAPE5=INPUT,TAPE6=
     1OUTPUT,TAPE8=DUMMY)
C
C     *** COMMON DECK ***
C
      COMMON/FEAT/IFTR(10)
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/PROP/COMPNT(8),APC(8),ATC(8),AVC(8),AMW(8),AOMEG(8),
     1 ADEL(8),AVW(8),APH(8),BET(8),GAM(8),DTA(8),BASEA(8),BASEB(8),
     2 ZCD(8),ALD(8)
      COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR
      COMMON/PATH/JPATH(8,200),NPATH(20),NPTHS
      COMMON/REFL/NLEV,RLEV(10)
C**   BLANK COMMON
      COMMON NNPR(2),NNSER(2),NNSM(2),SMCHA(8,50,2),SMCHB(7,50,2),
     1 SMCHX(8,10,2)
      COMMON NEMCH,EMCH(15,100)
C     ***
C
      INTEGER COMPNT
      DIMENSION TITLE(8),PROP(8,15),XX(8)
      EQUIVALENCE (PROP,APC)
C
      NAMELIST/PARLST/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR
      NAMELIST/COMP/NOCOMP,COMPNT
C
C     READ COMPONENT INFORMATION
      READ(5,COMP)
C     READ COMPONENT PHYSICAL CONSTANTS
      DO 10 I=1,NOCOMP
   10 READ(5,11)(PROP(I,K),K=1,15)
C
C     READ TITLE
      READ(5,100)TITLE
      IF(EOF,5)1,2
    1 CALL EXIT
    2 WRITE(6,101)TITLE
C     READ FEATURE CARD
      READ(5,105)IFTR
C     READ GENERAL SYSTEM PARAMETERS
      READ(5,PARLST)
C
C     READ SMCH A+B
      READ(5,105)NNPR,NNSER
      DO 50 K=1,2
      NNN=NNSM(K)=NNPR(K)
      DO 30 J=1,NNN
      READ(5,31)(SMCHA(L,J,K),L=1,8)
   30 WRITE(6,31)(SMCHA(L,J,K),L=1,8)
      DO 40 J=1,NNN
      READ(5,32)(SMCHB(L,J,K),L=1,7)
      WRITE(6,32)(SMCHB(L,J,K),L=1,7)
      READ(5,33)(XX(I),I=1,NOCOMP)
      FLOW=SMCHB(7,J,K)
      DO 38 I=1,NOCOMP
   38 SMCHX(I,J,K)=XX(I)/FLOW
   40 WRITE(6,33)(SMCHX(I,J,K),I=1,NOCOMP)
   50 CONTINUE
C
C     READ REFRIGERATION LEVELS
      READ(5,105)NLEV
      READ(5,33)(RLEV(I),I=1,NLEV)
      CALL SMATCH
C
   11 FORMAT(3(/5E14.5))
   31 FORMAT(5F6.0,5X,3F8.1)
   32 FORMAT(3F8.1,5X,F8.1,F10.0,F7.3,F8.1)
   33 FORMAT(7F10.4)
  100 FORMAT(8A10)
  101 FORMAT(8A10//)
  105 FORMAT(10I5)
      END
```

```
      SUBROUTINE SMATCH
C
C     COMPUTES ALL EQUIPMENT SEQUENCES TO SATISFY TEMPERATURE + PRESSURE
C     DEMANDS FOR A SET OF  HOT  +  COLD  STREAMS  (ICH=1 HOT , 2 COLD)
C
C     1. SATISFIES ALL PRESSURE CHANGE DEMANDS BY ADIABATIC EXPANSION
C     OR MULTISTAGE COMPRESSION (VAPOR ONLY)
C     2. COMPUTES ALL POSSIBLE MATCHES BETWEEN STREAMS , RESIDUALS
C     + SERVICE STREAMS (STEAM , WATER + REFR)
C
C     FEATURES ARE ACTIVATED BY 1 ENTRY IN /FEAT/
C                 1. STEAM
C                 2. SALE
C                 3. WATER
C                 4. REFRIGERATION
C                 5. VAPOR RECOMPRESSION
C
C     PRE-ASSIGNED RULES ARE USED TO PRE- SCREEN POSSIBLE MATCHES
C
C     SMCHA - STREAM CONTROL VECTORS -
C             1. PRIMARY STREAM NO
C             2. SECONDARY STREAM NO
C             3. ACTIVE/INACTIVE FLAG - 0. ACTIVE , 1. INACTIVE
C             4. STREAM TYPE - 1. FEED , 0. INTERMEDIATE , -1. PRODUCT
C               (2. HIGH PRIORITY - SATISFY BY SERVICE STREAM ONLY)
C               (-2. PSEUDO-SERVICE STREAM - PARALLEL PROCESSING)
C             5. STREAM SUB-TYPE - 0. LOAD , 1. (HEAT,REFR) SOURCE
C             6. FREE
C             7. PRESSURE SPEC
C             8. TEMP SPEC (-1. = FREE)
C
C     MATCH COUNT ARRAYS - MACC , MREJ
C        MACC - ACCEPTED
C                 1. STEAM
C                 2. SOLD
C                 3. WATER
C                 4. REFR
C                 5. PROCESS STREAM
C                 7. ADIABATIC EXPANSION
C                 8. COMPRESSION
C
C        MREJ - REJECTED - (2-5 APPLY ONLY TO PROCESS/PROCESS MATCHES)
C                 1. GENERAL TECH INFEASIBILITY
C                 2. VAP RECOMP - VRTMX EXCEEDED
C                 3. EXCEEDS MAX ENTROPY INCREASE/BTU  (DENMX)
C                    OR EXCEEDS MAX INLET TEMP DIFF (SUSP FOR HOT SOURCE)
C                 4. Q LT QMIN
C                 5. VAPOR/VAPOR MATCH
C                 6. FEED/INTERMEDIATE MATCH                              H
C
C     ****  COMMON
      COMMON/FEAT/IFTR(10)
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/EQUIP/EQUIP(15)
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
      COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR
      COMMON/PATH/JPATH(8,200),NPATH(20),NPTHS
      COMMON/REFL/NLEV,RLEV(10)
C**   BLANK COMMON
      COMMON NNPR(2),NNSER(2),NNSM(2),SMCHA(8,50,2),SMCHB(7,50,2),
     1 SMCHX(8,10,2)
      COMMON NEMCH,EMCH(15,100)
C     ****
      DIMENSION JIN(2),ICH(2)
      DIMENSION JCHA(2,6),NPR(2),NSS(2),JACT(2),JTP(2),JSTP(2)
      DIMENSION PSPEC(2),TSPEC(2),TEX(2),ISAT(2),DT(2)
      DIMENSION JHIST(20,2),NHS(2)
      DIMENSION MACC(10),MREJ(10)
      EQUIVALENCE (NPR,JCHA(1,1)),(NSS,JCHA(1,2)),(JACT,JCHA(1,3)),
     1 (JTP,JCHA(1,4)),(JSTP,JCHA(1,5))
C
      DATA PRMAX,RDTMX,VRTMX/4.,50.,60./
```

```
      DATA QMIN,TLIM/2.5E5,470./
      DATA DENMX,DTMAX/25.5,100./
C
C     INDEX DISPLACEMEMENT FUNCS FOR JPATH + NPATH ARRAYS
      DATA NPTHS/20/
      IDJ(JPR,JCH)=NPTHS*((JPR-1)+NNPR(1)*(JCH-1))
      IDN(JPR,JCH)=JPR+NNPR(1)*(JCH-1)
C
      NEMCH=0
      CALL ZERO(JPATH,1620)
      CALL ZEROI(MACC,20)
C*****************************************************************
C
C***  SATISFY ALL PRESSURE SPECIFICATIONS  ***
      WRITE(6,771)
      IPRES=1
C
      DO 50 KCH=1,2
      NNS=NNSM(KCH)
      DO 50 JS=1,NNS
C     LOAD STREAM JS INTO SIN(1)
      JIN(1)=JS
      ICH(1)=KCH
      NI=1
      ASSIGN 10 TO JLD
      GO TO 60
   10 PRES=PSPEC(1)
      IF(PRES.EQ.0.) PRES=PIN(1)
C     IF PRESSURE TO BE CHANGED , SET INACTIVE FLAG FOR INPUT STREAM
      IF(PRES.NE.PIN(1)) SMCHA(3,JS,KCH)=1.
      IF((PRES-PIN(1))) 15,50,20
C
C**   ADBIATIC FLASH
C
C     CALL ADBF , LOAD EQUIP + OUTPUT STREAM
   15 NACC=7
      ACTF=0.
      IOP=10
      JIN(2)=0
      ASSIGN 50 TO LOADSE
      GO TO 400
C
C**   COMPRESSION
C     GO TO COMPRESSION ROUTINE
   20 GO TO 450
C
   50 CONTINUE
      WRITE(6,772)
      GO TO 70
C*****************************************************************
C
C*    ROUTINE TO LOAD STREAM JS INTO INPUT NI (KCH=1 HOT , 2 COLD)
   60 DO 62 K=1,6
   62 JCHA(NI,K)=SMCHA(K,JS,KCH)
      PSPEC(NI)=SMCHA(7,JS,KCH)
      TEX(NI)=TSPEC(NI)=SMCHA(8,JS,KCH)
      KPR=NPR(NI)
      CALL MVFSM(NI,JS,KCH,KPR)
      GO TO JLD,(10,115)
C*********
C
C***  COMPUTE ALL POSSIBLE HEAT EXCHANGE MATCHES  ***
C
   70 IPRES=0
C     SET UP COUNTERS - SCAN ACROSS C FOR EACH H
      MIN1=-1
      NHO2=NCO2=0
  110 NNH2=NNSM(1)
      NNC2=NNSM(2)
C
      DO 201 NNH=MIN1,NNH2
      DO 200 NNC=MIN1,NNC2
C
C     HAS MATCH BEEN COMPUTED BEFORE
```

```
      IF(NNH.LE.NHO2.AND.NNC.LE.NCO2) GO TO 200
C
      ISERV=0
      ACTF=0.
C
C     LOAD STREAM CONTROL INFO INTO WORKING VECTORS
C     LOAD HOT INTO 1 , COLD INTO 2 (1 IF NNH LT 1)
      DO 120 KCH=1,2
      IF(KCH.EQ.1) JS=NNH
      IF(KCH.EQ.2) JS=NNC
      NI=KCH
      IF(NNH.LT.1) NI=3-KCH
      JIN(NI)=JS
      IF(JS.LT.1) GO TO 120
      ICH(NI)=KCH
      ASSIGN 115 TO JLD
      GO TO 60
C     TEST CONTROL INFORMATION
  115 IF(JACT(NI).EQ.1) GO TO 200
      IF(JTP(NI).EQ.-2) ISERV=NI
  120 CONTINUE
C
C-------------------------------
C---  SPECIFIC MATCH SELECTIONS/REJECTIONS ---
      IF(NNH.NE.1) GO TO 121                                            HP
      IF(NNC.EQ.2.OR.NNC.EQ.-1) GO TO 121                               HP
      GO TO 200                                                         HP
C-------------------------------
  121 IF(NNH.GT.0.AND.NNC.GT.0) GO TO 122
      IF(ISERV.GT.0) GO TO 200
      IF(NNH.LT.1) GO TO 125
      IF(NNC.LT.1) GO TO 140
C++++
C     *HEURISTIC* PRE-SCREENING TO REJECT TYPE-INFEASIBLE MATCHES
C*    DISALLOW SOURCE/SOURCE MATCHES
  122 IF((JSTP(1)*JSTP(2)).EQ.1) GO TO 200
C*    DISALLOW VAPOR/VAPOR MATCHES
      IF((VFIN(1)*VFIN(2)).EQ.1) GO TO 210
C*    DISALLOW FEED/INTERMEDIATE MATCHES                                HP
      DO 123 J=1,2                                                      HP
  123 IF(JTP(J).EQ.1.AND.JTP(3-J).EQ.0) GO TO 212                       HP
C*    FOR JTP( )=2 SAT BY SERVICES ONLY
      IF(JTP(1).EQ.2.AND.NNC.GT.0) GO TO 200
      IF(JTP(2).EQ.2.AND.NNH.GT.-1) GO TO 200
      GO TO 156
C++++
C**   COLD STREAM *NNC*  - MATCHES WITH  STEAM(NNH=-1) , VALUE(NNH=0)
C
  125 IF(NNH.EQ.0) GO TO 130
C*    STEAM - INVALID IF SOURCE
      IF(IFTR(1).EQ.0) GO TO 200
      IF(JSTP(1).EQ.1) GO TO 200
C     CALL HXER + LOAD EQUIP
      NACC=1
      IOP=2
      IS2=1
      ASSIGN 200 TO LOADSE
      GO TO 400

C*    SALE - VALID FOR SOURCE ONLY
  130 IF(IFTR(2).EQ.0) GO TO 200
      IF(JSTP(1).NE.1) GO TO 200
      NACC=2
      IOP=30
      IS2=0
      ASSIGN 200 TO LOADSE
      GO TO 400
C
C**   HOT STREAM *NNH* - MATCHES WITH  WATER(NNC=-1) , REFR(NNC=0)
C     - INVALID FOR SOURCE
C
  140 IF(JSTP(1).EQ.1.AND.NPR(1).NE.1) GO TO 200                        HP
      TWA=TWAT+APPP
      IOP=1
```

```
      IF(NNC.EQ.0) GO TO 150
C*    WATER
      IF(IFTR(3).EQ.0) GO TO 200
      IF(TIN(1).LT.(TWA+DTW)) GO TO 202
      IS2=2
      IF(TSPEC(1).GT.TWA) GO TO 142
C     CAN ONLY COOL TO TWAT+APPP
      TEX(1)=TWA
C     CALL HXER + LOAD EQUIP
  142 NACC=3
      ASSIGN 200 TO LOADSE
      GO TO 400
C
C*    REFR
C     SET STREAM OUTLET TEMP
C     SET UP RESIDUAL IF MAX TEMP CHANGE (RDTMX) IS EXCEEDED
C     REJECT IF CAN BE PARTIALLY SAT BY WATER
  150 IF(IFTR(4).EQ.0) GO TO 200
      IF(TIN(1).GT.(TWA+DTW)) GO TO 200
      IS2=3
C     IS RDTMX EXCEEDED
      IF((TIN(1)-TEX(1)).LT.RDTMX) GO TO 154
      TEX(1)=TIN(1)-RDTMX
C     IF WITHIN 10 DEG OF AVAIL LEVEL , COOL ONLY TO THIS LEVEL
      DO 152 I=1,NLEV
  152 IF(ABS(TEX(1)-RLEV(I)).LT.10.) TEX(1)=RLEV(I)+APRR
C     MAKE EQUIP ENTRY + LOAD RESID IF NECC
  154 NACC=4
      ASSIGN 200 TO LOADSE
      GO TO 400
C
C***  PROCESS STREAM MATCH BETWEEN NNC,NNH
C     CONSTRAINTS - ( SEE MREJ ARRAY)
C
  156 IOK=1
C*    CHECK FOR INFEAS DUE TO MULTIPLE STREAM USAGE
C     - EXCEPT FOR PSEUDO-SERVICE STREAM
      IF(ISERV.GT.0) GO TO 162
C     RECOVER STREAM HISTORIES
      DO 158 J=1,2
      JST=JIN(J)*(3-2*ICH(J))
  158 CALL SHIST(JST,NHS(J),JHIST(1,J))
C     CHECK HISTORIES FOR COMMON STREAMS
      I1=NHS(1)
      I2=NHS(2)
      DO 160 I=1,I1
      DO 160 J=1,I2
  160 IF(JHIST(I,1).EQ.JHIST(J,2)) GO TO 200
C
C*    TEST INLET TEMPS
  162 APMIN=APRR
      IF(TIN(2).LT.TRRR) APMIN=ARRR
      DTIN=TIN(1)-TIN(2)
C     CHECK FOR DTIN GT MAX (EXCEPT FOR HOT SOURCE)
      IF(DTIN.GT.DTMAX.AND.JSTP(1).EQ.0) GO TO 206
C     IS VAP RECOMP A POSSIBILITY (IOK=0)
      IF(DTIN.LT.APMIN.AND.ISERV.EQ.0) IOK=0
C
C     COMPUTE DTS - IF ONE TSPEC FREE SET TO GIVE MIN APPROACH
      DO 164 J=1,2
      JJ=3-J
      SIGN=3-2*J
      IF(IOK.EQ.0) GO TO 164
      IF(TSPEC(JJ).EQ.-1.) TEX(JJ)=TIN(J)-SIGN*APMIN
  164 DT(J)=SIGN*(TIN(J)-TEX(JJ))
      IF(IOK.EQ.0) GO TO 170
C
C     COMPUTE CLOSEST APPROACH + CHECK WHETHER GT MIN
      CLAP=AMIN1(DT(1),DT(2))
      IF(CLAP.GT.APMIN) GO TO 168
C     LOWER BOUND IS VIOLATED - CAN ONLY SAT TEMP SEG BY EX TO APPROACH
C     - REJECT FOR PSEUDO-SERVICE
      IF(ISERV.GT.0) GO TO 202
C     SET UP APPROACH TEMPS
```

```
      DO 166 J=1,2
      SIGN=3-2*J
  166 IF(DT(J).LT.APMIN) TEX(3-J)=TIN(J)-SIGN*APMIN
  168 NACC=5
      IOP=1
      IS2=4+ISERV
C*    CALL HXER , LOAD EQUIP , TEST FOR RESIDUALS + LOAD OUTPUT STREAMS
      ASSIGN 200 TO LOADSE
      GO TO 400
C
C**   VAPOR RECOMPRESSION
C
  170 IF(IFTR(5).EQ.0) GO TO 200
C++++
C*    ARE BOTH PRIMARY STREAMS - IS EITHER A SOURCE
C     -HAS EITHER STREAM UNDERGONE PRES CHANGE
      DO 171 J=1,2
      IF(NSS(J).GT.0) GO TO 202
      IF(JSTP(J).EQ.1) GO TO 202
  171 IF(PSPEC(J).GT.0.) GO TO 202
C     CAN HOT STREAM BE SATISFIED BY WATER
      IF(TSPEC(1).GT.TWA) GO TO 200
C++++
C
C*    REJECT IF PHASE CHANGES NOT POSSIBLE FOR BOTH STREAMS
      DO 250 J=1,2
      TI=TIN(J)
      TX=TSPEC(J)
      BP=BPIN(J)
      DP=DPIN(J)
      IF(TI.LE.BP.AND.TX.LE.BP) GO TO 202
  250 IF(TI.GE.DP.AND.TX.GE.DP) GO TO 202
C
C*    APPROX TEMP DIFFERENCE TEST
      IF(AMAX1(-DT(1),-DT(2)).GT.VRTMX) GO TO 204
C*    TEST FOR PRESSURE RATIO TO ACHIEVE MIN APPROACH AT COLD EXIT
      TSAV=TIN(1)
      PSAV=PIN(1)
      TIN(1)=TSPEC(2)+APRR
      PIN(1)=2.0*PIN(1)
      CALL PDEW(1,PRES,DUM)
      WRITE(6,760)NNH,NNC,PSAV,PRES
      TIN(1)=TSAV
      PIN(1)=PSAV
C     PRES=0. INDICATES ABOVE CRITICAL PRESSURE
      IF(PRES.EQ.0.) GO TO 204
C++   ALLOW 0.10 FRAC INCREASE IN VAP FLOW DUE TO FLASH-OFF ON COL RE-ENT
      CALL SPLIT(1.10)
      CALL MVSOSI(1,1,0,0)
C*    GO TO COMPRESSION ROUTINE
      GO TO 450
C     SET NEW TEMP SPEC FOR OUTLET STREAM
  172 TSPEC(1)=TEX(1)=BPIN(1)
      JIN(2)=NNC
C*    EXCHANGE
      GO TO 168
C
C
C**   COUNT REJECTIONS BY CATEGORY
  202 NREJ=1
      GO TO 220
  204 NREJ=2
      GO TO 220
  206 NREJ=3
      GO TO 220
  208 NREJ=4
      GO TO 220
  210 NREJ=5
      GO TO 220
  212 NREJ=6
  220 MREJ(NREJ)=MREJ(NREJ)+1
      WRITE(6,773)JIN,NREJ
  200 CONTINUE
  201 CONTINUE
```

```
C*****************************************************************
C
C      COMPUTE NO OF NEW RESIDUALS
C
       NRSH=NNSM(1)-NNH2
       NRSC=NNSM(2)-NNC2
       WRITE(6,774)MACC,MREJ,NNSM(1),NNSM(2),NRSH,NRSC,NEMCH
       IF((NRSH+NRSC).EQ.0) GO TO 670
C      SAVE OLD COUNTERS , SET NEW COUNTERS + COMPUTE FOR NEW RESIDUALS
       NCO2=NNC2
       NHO2=NNH2
       GO TO 110
C***************************
C
   670 WRITE(6,660)
       WRITE(6,600)NNPR,NNSER,NNSM
       WRITE(8,600)NNPR,NNSER,NNSM
       DO 625 JCH=1,2
       NNN=NNPR(JCH)
       DO 610 J=1,NNN
       WRITE(6,612)(SMCHX(II,J,JCH),II=1,NOCOMP)
   610 WRITE(8,612)(SMCHX(II,J,JCH),II=1,NOCOMP)
       WRITE(6,660)
       NNN=NNSM(JCH)
       DO 620 J=1,NNN
       WRITE(6,622)(SMCHA(II,J,JCH),II=1,8),J
       WRITE(8,622)(SMCHA(II,J,JCH),II=1,8)
       WRITE(6,624)(SMCHB(II,J,JCH),II=1,7)
   620 WRITE(8,624)(SMCHB(II,J,JCH),II=1,7)
   625 WRITE(6,660)
C
       DO 645 JCH=1,2
       JP1=IDJ(1,JCH)+1
       JP2=IDJ(NNPR(JCH),JCH)+NPTHS
       NP1=IDN(1,JCH)
       NP2=IDN(NNPR(JCH),JCH)
       WRITE(6,600)(NPATH(II),II=NP1,NP2)
       WRITE(6,660)
       WRITE(8,600)(NPATH(II),II=NP1,NP2)
       DO 640 JP=JP1,JP2
       XJP=FLOAT(JP-1)/FLOAT(NPTHS)
       YJP=INT(XJP)
       IF((YJP-XJP).EQ.0.) WRITE(6,660)
       WRITE(6,600)(JPATH(II,JP),II=1,6)
   640 WRITE(8,600)(JPATH(II,JP),II=1,6)
   645 WRITE(6,660)
       WRITE(6,600)NEMCH
       WRITE(8,600)NEMCH
       DO 650 K=1,NEMCH
       WRITE(6,652)(EMCH(II,K),II=1,15)
       WRITE(6,660)
   650 WRITE(8,652)(EMCH(II,K),II=1,15)
       RETURN
C
C***************************
C
C**    ROUTINE TO LOAD EQUIP NO NEMCH INTO STREAM PATH ARRAY   **
C      FOR STREAM JN (PARAMS JCH,JPR,JSS)
C      NCJ IS COL DISPLACEMENT FOR PRIM STREAM CORR TO JN
C      ROW 1 OF JPATH ARRAY CONTAINS NO OF EQUIP ENTRIES IN COL
C
   300 STR=(JN*(3-2*JCH))
       NCJ=IDJ(JPR,JCH)
       NROW=JSS+1
       IF(IPRES.EQ.0) GO TO 308
C
C      MAKE ENTRY IN COL 1
       NCOL=NCJ+1
       NROW=JPATH(1,NCOL)+1
       GO TO 330
C
C      IS STREAM NON-PRIMARY
   308 IF(NROW.NE.1) GO TO 310
C      CREATE NEW COL
```

```
      JC=2
      GO TO 315
C
C     LOCATE EQUIP FOR WHICH STREAM STR IS AN * OUTPUT *
C     SCAN ROW NROW
  310 DO 312 JC=2,NPTHS
      NEQ=JPATH(NROW,JC+NCJ)
      IF(NEQ.EQ.0) GO TO 312
      IF(EMCH(5,NEQ).EQ.STR.OR.EMCH(6,NEQ).EQ.STR) GO TO 314
  312 CONTINUE
  314 NCOL=NCL=JC+NCJ
C     IS NEXT ENTRY IN COL FREE
      IF(JPATH(NROW+1,NCOL).EQ.0) GO TO 330
C
C     CREATE NEW COL - FIRST FREE COL
  315 NP=IDN(JPR,JCH)
      NPATH(NP)=NPATH(NP)+1
      DO 316 KC=JC,NPTHS
  316 IF(JPATH(1,KC+NCJ).EQ.0) GO TO 318
      WRITE(6,777)JPR,JCH
      CALL EXIT
  318 NCOL=KC+NCJ
      JPATH(1,NCOL)=NROW
      IF(NROW.EQ.1) GO TO 330
      DO 320 NR=2,NROW
  320 JPATH(NR,NCOL)=JPATH(NR,NCL)
C
C*    ADD NEW EQUIP NO TO COL - IN NROW+1
  330 JPATH(1,NCOL)=NROW
      JPATH(NROW+1,NCOL)=NEMCH
      WRITE(6,776)NCOL,NROW,(JPATH(NN,NCOL),NN=1,6)
      GO TO 504
C***********************************************************************
C
C***  EQUIP CALLING ROUTINE  **
C
C     INCREMENT MATCH COUNTERS
  400 MACC(NACC)=MACC(NACC)+1
      JNI=2
      IF(JIN(2).LE.0) JNI=1
      WRITE(6,700)JIN,IOP
      DO 706 J=1,JNI
      WRITE(6,622)(JCHA(J,II),II=1,5),PSPEC(J),TSPEC(J)
  706 WRITE(6,624)BPIN(J),DPIN(J),TIN(J),PIN(J),HIN(J),VFIN(J),TMIN(J)
C
      IF(IOP.EQ.10) GO TO 401
      IF(IOP.EQ.11) GO TO 410
      IF(IOP.LE.3) GO TO 420
      IF(IOP.EQ.30) GO TO 480
C
C**   ADIABATIC FLASH **
  401 NOUT=NIN=1
      CALL ADBF(PRES)
      GO TO 500
C
C**   COMPRESSOR **
  410 CALL COMP(PRES)
      GO TO 500
C
C**   HEAT EXCHANGER **
  420 IF(IS2.LE.3) JIN(2)=IS2+200
      IF(IS2.EQ.3) GO TO 480
      CALL HXER(IOP,IS2,TEX,Q)
      IF(IS2.LT.4) GO TO 500
      IF(ABS(Q).LT.QMIN) GO TO 212
C*    ENTROPY INCREASE TEST FOR PROC/PROC MATCH (NOT FOR HOT SOURCE)
      IF(EQUIP(9).LT.DENMX.OR.JSTP(1).EQ.1) GO TO 500
      WRITE(6,778)EQUIP(9)
      GO TO 206
C
C***  (MULTI-STAGE) COMPRESSION ROUTINE - WATER INTERCOOLING IF REQD
C     COMPUTE NO OF STAGES - EQUAL PRES RATIO/STAGE
  450 PR=PRES/PIN(1)
      STAGE=ALOG(PR)/ALOG(PRMAX)
```

```
      NSTG=INT(STAGE+0.999)
      PRSTG=PR**(1./FLOAT(NSTG))
C
      DO 460 KK=1,NSTG
      ACTF=1.
      PRES=PIN(1)*PRSTG
C     CALL COMPR , LOAD EQUIP + OUTPUT STREAM
      NACC=8
      IOP=11
      JIN(2)=0
      ASSIGN 452 TO LOADSE
      GO TO 400
  452 JIN(1)=NS
C     NO AFTERCOOLING FOR IPRES=0 , KK=NSTG
      IF(KK.EQ.NSTG.AND.IPRES.EQ.0) GO TO 460
C     AFTERCOOL , LOAD STREAM + EQUIP
      IF(TIN(1).LT.(TWAT+APPP+DTW)) GO TO 460
      NACC=3
      IF(KK.LT.NSTG) ACTF=1.
      IOP=1
      IS2=2
      TEX(1)=AMAX1((TWAT+APPP),(DPIN(1)+1.))
      ASSIGN 460 TO LOADSE
      GO TO 400
  460 JIN(1)=NS
      IF(IPRES.EQ.1) GO TO 50
      IF(IPRES.EQ.0) GO TO 172
C
C***  EQUIP ENTRY ONLY - UNLESS RESIDUAL IS INDICATED
C     EQUIP TYPES - (1. REFR) , (30. SALE)
  480 CALL ZERO(EQUIP,15)
      EQUIP(2)=IOP
      IF(IOP.EQ.30) GO TO 482
      EQUIP(10)=TOUT(1)=TIN(1)=TEX(1)
C     COMPUTE OUTLET STREAM CONDITION IF TSPEC NOT MET
      NIN=NOUT=1
      IF(TEX(1).NE.TSPEC(1)) CALL ISOF(0.)
      GO TO 500
C
  482 EQUIP(7)=TIN(1)
      EQUIP(8)=TEX(1)
      GO TO 500
C****************************
C
C***  EQUIP LOADING ROUTINE  **
C
  500 EQUIP(1)=NEMCH=NEMCH+1
      ISIGN1=3-2*ICH(1)
      EQUIP(3)=JIN(1)*ISIGN1
      EQUIP(4)=JIN(2)
      IF(JNI.EQ.2) EQUIP(4)=JIN(2)*(-ISIGN1)
      WRITE(6,662)
      IF(IPRES.EQ.1) GO TO 502
C     FOR PSEUDO-SERVICE STREAM ADD 300. TO STREAM CODE
      IF(ISERV.GT.0) EQUIP(ISERV+2)=EQUIP(ISERV+2)+300.
C
C*    MAKE ENTRIES IN STREAM PATH ARRAY - FOR INPUT STREAMS
C     - EXCEPT PSEUDO-SERVICES
  502 DO 504 J=1,JNI
      IF(JTP(J).EQ.-2) GO TO 504
      JN=JIN(J)
      JCH=ICH(J)
      JPR=NPR(J)
      JSS=NSS(J)
      GO TO 300
  504 CONTINUE
      IF(IOP.EQ.30) GO TO 540
C
C*    CHECK TEMP SPECS
      DO 510 J=1,JNI
      ISAT(J)=0
      IF(IOP.GT.3) GO TO 510
      SPECT=TSPEC(J)
      IF(SPECT.EQ.-1.) SPECT=TLIM
```

```
      SIGN=2*ICH(J)-3
      TEST=(TOUT(J)-SPECT)*SIGN
      IF(TEST.GT.-0.01) ISAT(J)=1
  510 CONTINUE
C
C*    LOAD OUTPUTS INTO NEW STREAM LOCATIONS
      DO 530 NO=1,JNI
      IF(ISAT(NO).EQ.1) GO TO 530
      JCH=ICH(NO)
      JPR=NPR(NO)
C
      NS=NNSM(JCH)=NNSM(JCH)+1
      IF(IPRES.EQ.1) GO TO 520
C     INCR SECONDARY STREAM NO
      NSS(NO)=NSS(NO)+1
C     SET REMAINING CONTROL INFORMATION FOR NEW STREAM
C*
C++   IF ACT,INT,LOAD SET TYPE TO 2                                        H
      IF(ACTF.EQ.0..AND.JTP(NO).EQ.0.AND.JSTP(NO).EQ.0) JTP(NO)=2          H
C*
  520 JACT(NO)=ACTF
      ACTF=0.
      DO 522 K=1,6
  522 SMCHA(K,NS,JCH)=JCHA(NO,K)
      SMCHA(7,NS,JCH)=PSPEC(NO)
      SMCHA(8,NS,JCH)=TSPEC(NO)
C     LOAD STREAM PROPS
      CALL MVTSM(-NO,NS,JCH,JPR)
C     FOR NO=1,JNI=1 RESTORE OUTPUT TO INPUT FOR FURTHER PROCESSING
      IF((NO*JNI).EQ.1) CALL MVSOSI(1,1,0,0)
      WRITE(6,750)
      WRITE(6,622)(SMCHA(II,NS,JCH),II=1,8)
      WRITE(6,624)(SMCHB(II,NS,JCH),II=1,7)
      EQUIP(NO+4)=NS*(3-2*JCH)
  530 CONTINUE
C
C*    LOAD EQUIP INFORMATION
  540 DO 542 K=1,15
  542 EMCH(K,NEMCH)=EQUIP(K)
      WRITE(6,770)EQUIP
      GO TO LOADSE,(50,200,452,460)
C
  600 FORMAT(10I5)
  612 FORMAT(7F10.4)
  622 FORMAT(5F6.0,5X,3F8.1,I7)
  624 FORMAT(3F8.1,5X,F8.1,F10.0,F7.3,F8.1)
  652 FORMAT(3(2F5.0,3X),F12.0/5F9.1,3F9.0)
  660 FORMAT(//)
  700 FORMAT(/1H ,12(1H*)/2H *,2I4,3H  *,I5/1H ,12(1H*))
  750 FORMAT(/* ++ LOADED OUTPUT-*)
  760 FORMAT(//* -- VAP RECOMP - NNH,NNC*,2I5,* PIN,PRES*,2F8.1//)
  770 FORMAT(/* ..EQUIP*/3(2F5.0,3X),F12.0,5F9.1,3F9.0/)
  771 FORMAT(//// * +++++ PRESSURE SPECS +++++*//)
  772 FORMAT(///* +++++ TEMPERATURE SPECS +++++*//)
  773 FORMAT(* JIN,NREJ*,2I4,5X,I4)
  774 FORMAT(//* MACC*,10I5/* MREJ*,10I5//* NSMH,NSMC,NRSH,NRSC*,
     1 2I5,10X,2I5//* NEMCH*,I5/1H1)
  776 FORMAT(* NCOL,NROW,COL*,2I6,5X,6I5)
  777 FORMAT(//1H ,20(1H*),* NPTHS EXCEEDED , JPR,JCH=*,2I5)
  778 FORMAT(* DENT*,F8.1)
      END
```

```
      SUBROUTINE STMOVS(IWV,ISM,III,NX)
C
C     STREAM MOVING UTILITY ROUTINE ...(SMATCH + BRBND VERSION)
C
C     IWV   - ELEMENT NUMBER IN SIN OR SOUT WORKING ARRAY
C             + SIN
C             - SOUT
C     ISM   - VECTOR NUMBER IN SM--
C     III   - 0     MOVE TO OR FROM SMPB
C             1-2   MOVE TO OR FROM SMCHB - (1-2)
C             3-5   MOVE TO OR FROM SMRB -  (1-3)
C     NX    - VECTOR NUMBER CONTAINING MOLE FRACTIONS   (III GT 0)
C
C     3 ENTRIES -
C       1 MVSOSI MOVES SOUT VECTOR ISM TO SIN VECTOR IWV (III=0)
C       2 MVFSM MOVES FROM SM-- TO SIN OR SOUT
C       3 MVTSM MOVES TO SM-- FROM SIN OR SOUT
C
C     *** COMMON DECK ***
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
C**   BLANK COMMON
      COMMON NNPR(2),NNSER(2),NNSM(2),SMCHA(8,50,2),SMCHB(7,50,2),
     1 SMCHX(8,10,2)
C     ***
      DIMENSION SMPB(1,1,1),SMRB(1,1,1),SMRX(1,1,1)
      DIMENSION SIDUM(4,7),SODUM(4,7)
      EQUIVALENCE (BPIN,SIDUM),(BPOUT,SODUM)
C
C**
      ENTRY MVSOSI
      IENT=1
      GO TO 1
C**
      ENTRY MVFSM
      IENT=2
      GO TO 1
C**
      ENTRY MVTSM
      IENT=3
C
    1 JJJ=III+1
      GO TO (2,3,3,4,4)JJJ
    2 ITYPE=1
      GO TO 5
    3 ITYPE=2
      KKK=III
      GO TO 5
    4 ITYPE=3
      KKK=III-2
C
    5 GO TO (100,200,300)IENT
C
C**   MVSOSI
  100 DO 50 I=1,7
   50 SIDUM(IWV,I)=SODUM(ISM,I)
      DO 60 I=1,NOCOMP
   60 XIN(I,IWV)=XOUT(I,ISM)
      RETURN
C
C**   MVFSM
  200 DO 10 I=1,7
      GO TO (6,7,8)ITYPE
    6 AA=SMPB(I,ISM)
      GO TO 9
    7 AA=SMCHB(I,ISM,KKK)
      GO TO 9
    8 AA=SMRB(I,ISM,KKK)
    9 IF(IWV.GT.0) SIDUM(IWV,I)=AA
      IF(IWV.LT.0) SODUM(-IWV,I)=AA
   10 CONTINUE
C
```

```
      DO 20 I=1,NOCOMP
      GO TO (11,12,13)ITYPE
   11 AA=SMPB(I+7,ISM)
      GO TO 18
   12 AA=SMCHX(I,NX,KKK)*SMCHB(7,ISM,KKK)
      GO TO 18
   13 AA=SMRX(I,NX,KKK)*SMRB(7,ISM,KKK)
   18 IF(IWV.GT.0) XIN(I,IWV)=AA
      IF(IWV.LT.0) XOUT(I,-IWV)=AA
   20 CONTINUE
      RETURN
C
C**   MVTSM
  300 DO 30 I=1,7
      IF(IWV.GT.0) AA=SIDUM(IWV,I)
      IF(IWV.LT.0) AA=SODUM(-IWV,I)
      GO TO (21,22,23)ITYPE
   21 SMPB(I,ISM)=AA
      GO TO 30
   22 SMCHB(I,ISM,KKK)=AA
      GO TO 30
   23 SMRB(I,ISM,KKK)=AA
   30 CONTINUE
C
      IF(ITYPE.GT.1) RETURN
      DO 40 I=1,NOCOMP
      IF(IWV.GT.0) AA=XIN(I,IWV)
      IF(IWV.LT.0) AA=XOUT(I,-IWV)
   40 SMPB(I+7,ISM)=AA
      RETURN
      END
```

```
      SUBROUTINE SHIST(JSS,NHIST,JHIST)
C
C     ROUTINE GENERATES STREAM HISTORY FOR STREAM JSS
C     - COMPILES LIST OF STREAMS (JHIST,NO. NHIST) USED IN PRODUCING JSS
C
      COMMON/PATH/JPATH(8,200),NPATH(20),NPTHS
      COMMON NNPR(2),NNSER(2),NNSM(2),SMCHA(8,50,2),SMCHB(7,50,2),
     1 SMCHX(8,10,2)
      COMMON NEMCH,EMCH(15,100)
C
      DIMENSION JHIST(1),JIS(10)
C
C     INDEX DISPLACEMENT FUNCS FOR JPATH , NPATH ARRAYS
      IDJ(JPR,JCH)=NPTHS*((JPR-1)+NNPR(1)*(JCH-1))
      IDN(JPR,JCH)=JPR+NNPR(1)*(JCH-1)
C
      JJ=NIS=0
      NHIST=1
      JHIST(1)=JSS
      CALL ZEROI(JIS,10)
      JSTR=JSS
      GO TO 12
C
C     SELECT NEXT STREAM FROM JIS
   10 JJ=JJ+1
      IF(JJ.GT.NIS) RETURN
      JSTR=JIS(JJ)
   12 JCH=(3-ISIGN(1,JSTR))/2
      JSR=IABS(JSTR)
      JPR=SMCHA(1,JSR,JCH)
      JSEC=SMCHA(2,JSR,JCH)
      IF(NHIST.EQ.1.AND.JSEC.EQ.0) RETURN
      IF(JSEC.EQ.0) GO TO 10
C
C     LOCATE EQUIP NODE FROM WHICH JSTR IS AN OUTPUT
      NC1=IDJ(JPR,JCH)+1
      NCN=IDN(JPR,JCH)
      NC2=NC1+NPATH(NCN)
      NR=JSEC+1
      STR=JSTR
C
      DO 20 NCOL=NC1,NC2
      NEQ=JPATH(NR,NCOL)
      IF(NEQ.EQ.0) GO TO 20
      IF(EMCH(5,NEQ).EQ.STR.OR.EMCH(6,NEQ).EQ.STR) GO TO 22
   20 CONTINUE
C     SCAN UP REMAINDER OF COL NCOL SAVING INPUTS
   22 DO 30 N=2,NR
      NN=NR-N+2
      NEQ=JPATH(NN,NCOL)
      DO 26 NI=3,4
      NS=EMCH(NI,NEQ)
      IF(NS.EQ.0.OR.NS.GT.200) GO TO 26
C     IF NS IS A *SIDE STREAM* - OPP SIGN FORM JSTR , ADD TO JIS
      IF((NS*JSTR).GT.0) GO TO 24
      NIS=NIS+1
      JIS(NIS)=NS
C     ADD INPUTS TO JHIST
   24 NHIST=NHIST+1
      JHIST(NHIST)=NS
   26 CONTINUE
   30 CONTINUE
      GO TO 10
      END
```

C SELECTION OF OPTIMAL NETWORK CONFIGURATION (Branch and Bound Optimization)

MAINB reads the input data for this section. The data consists mainly of the stream, stream processing path and equipment arrays from the preceding section. The routine also sets up the stream energy cost splines from supplied temperature level/cost data.

ENERGY has two entries, ENEC called prior to BRBND and ENDS called immediately after BRBND.

ENEC is responsible for computing capital and operating costs for all equipment which involve energy costs, i.e. refrigeration exchangers and stream sales. This step completes the equipment costing process thus allowing costs to be summed for each complete processing path. The set of these paths for each primary stream is then sorted into order of increasing cost for convenience in the branch and bound optimization calculations.

After the optimal network configuration has been selected by BRBND, entry ENDS is accessed to compile lists of energy usages and transfers for the optimal plant, i.e. refrigeration demands, stream sales and pseudo-service usages.

BRBND is the branch and bound optimizing routine. Its task is to select the lowest cost feasible combination of stream processing paths which jointly define the optimal network configuration. The branch and bound procedure has been described earlier, in section 2.1, so that only a graphical algorithm for the actual routine is presented here, in Figure II.2. There are several notes of explanation which should be given.

i) The first is the two-pass solution method (IPAS = 1, 2). The first pass is used to establish a good feasible network whose cost provides a useful initial upper bound for the normal calculation path on the second pass. Only one level of branching is used for the first pass. This results in the evolution of a sufficient number of feasible networks to produce a good bound without necessitating an excessive computation time for the procedure. On the second pass branching continues down to the number of levels specified, at which point problems are solved as indicated in Figure II.2.

ii) The basic algorithms for, (a) establishing good feasible networks and for, (b) using their (bounding) costs to reject all processing paths which must lead to higher cost networks have been described by Lee et al[1] and will not be detailed here. However it should be pointed out that both procedures basically depend on having the set of processing paths for each primary stream sorted into increasing order of cost. This allows easy selection of either the lowest cost active path for any stream or the lowest cost active path which is compatible with a partial set of other paths already selected. More detail should be obtained from the reference given above and the actual program listing.

iii) Each processing path has an active/inactive flag which is conveniently used to indicate whether or not a path is active for the current problem. Paths are inactivated either through imcompatibility with bounding problems or because they must lead to networks of higher cost then the present bound. The branching structure dictates that the inactive flags retain information on the branching level at which paths were inactivated in order to be readily able to re-activate them at an appropriate point for succeeding problems. Thus inactive flags take the value of the level number at which they were set.

iv) The final point refers to the method of encoding path combinations (networks) into "plant numbers" used in the routine (but not indicated in Figure II.2). These numbers are needed for reference purposes. They have as digits the sequence numbers (1...) of the component stream processing paths which define the network in question. The number base is the maximum number of paths allocated per primary stream (NPTHS in the program). For an example refer to page 240. This scheme allows any process configuration to be encoded into a single number; decoding to identify component paths is accomplished by the reverse of the encoding procedure.

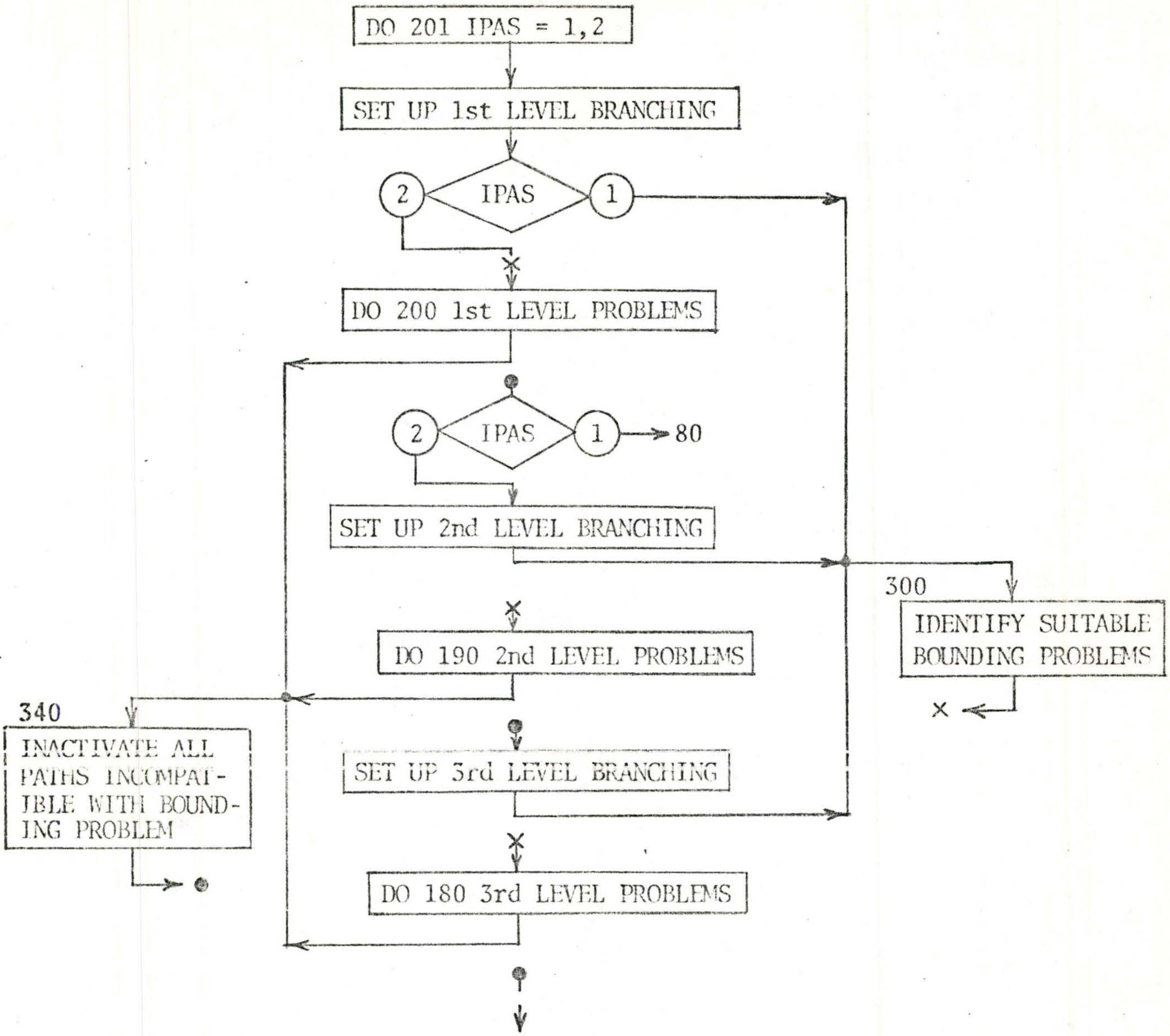


Figure II.2 BRBND Algorithm (Continued on page 148).

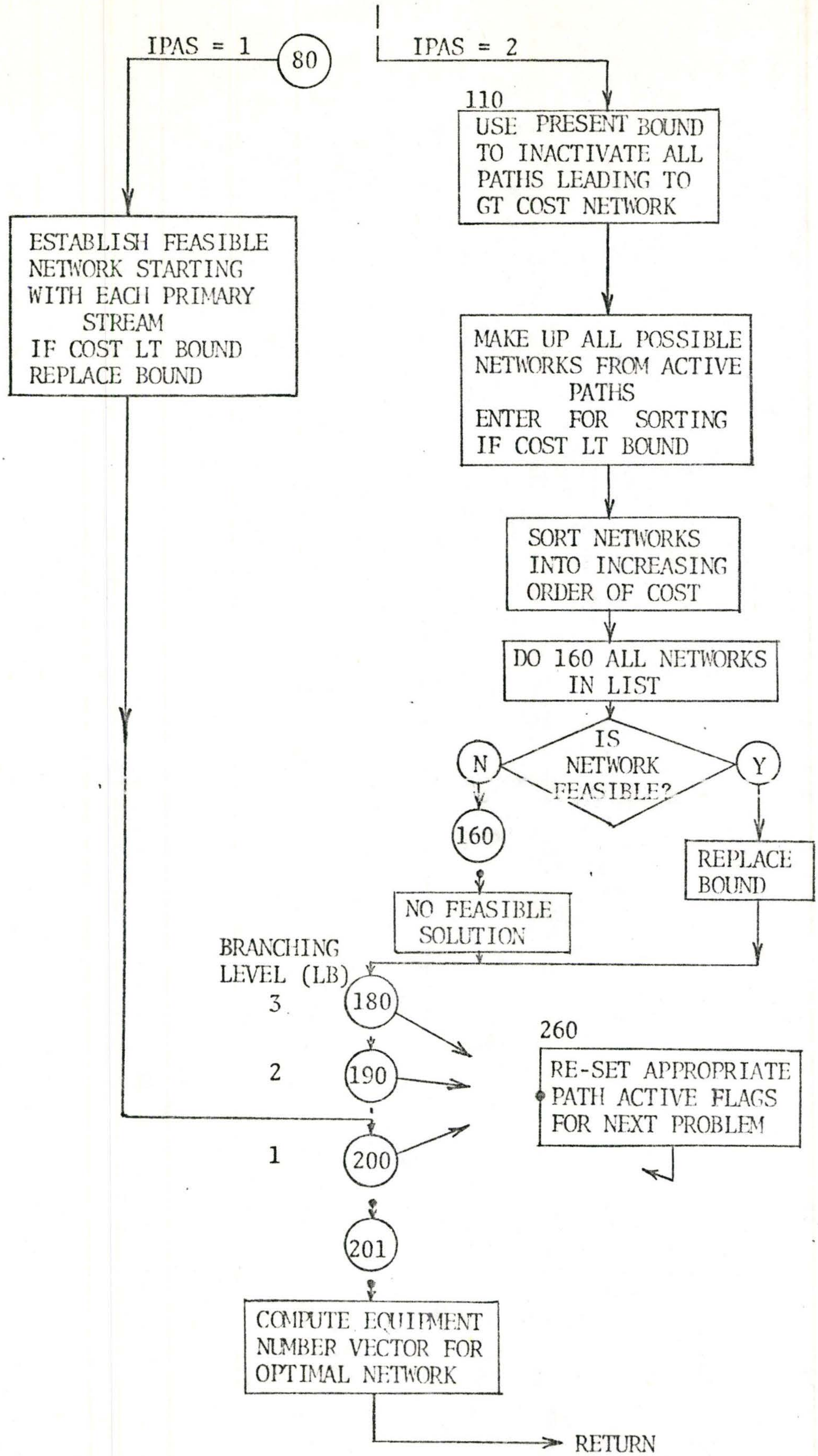


Figure II.2 BRBND Algorithm (Continued).

```
      PROGRAM MAINB(INPUT=1001,OUTPUT=1001,TAPE5=INPUT,TAPE6=OUTPUT,TAPE
     18=INPUT)
C
C     *** COMMON DECK ***
C
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/PROP/COMPNT(8),APC(8),ATC(8),AVC(8),AMW(8),AOMEG(8),
     1 ADEL(8),AVW(8),APH(8),BET(8),GAM(8),DTA(8),BASEA(8),BASEB(8),
     2 ZCD(8),ALD(8)
      COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR,DTF(2)
      COMMON/SPLINE/NH,NC,X(10),Y(10),PM(10)
      COMMON/PATH/JPATH(8,200),NPATH(20),NPTHS
C**   BLANK COMMON
      COMMON NNPR(2),NNSER(2),NNSM(2),SMCHA(8,50,2),SMCHB(7,50,2),
     1 SMCHX(8,10,2)
      COMMON NEMCH,EMCH(15,100)
C         ***
C
      INTEGER COMPNT
      DIMENSION TITLE(8),PROP(8,15)
      EQUIVALENCE (PROP,APC)
C
      NAMELIST/PARLST/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR,DTF
      NAMELIST/COMP/NOCOMP,COMPNT
C
C     INDEX DISPLACEMEMENT FUNCS FOR JPATH + NPATH ARRAYS
      IDJ(JPR,JCH)=NPTHS*((JPR-1)+NNPR(1)*(JCH-1))
      IDN(JPR,JCH)=JPR+NNPR(1)*(JCH-1)
C
C     READ COMPONENT INFORMATION
      READ(8,COMP)
C     READ COMPONENT PHYSICAL CONSTANTS
      DO 10 I=1,NOCOMP
   10 READ(8,11)(PROP(I,K),K=1,15)
C
C     READ TITLE
      READ(5,100)TITLE
      IF(EOF,5)1,2
    1 CALL EXIT
    2 WRITE(6,101)TITLE
C     READ GENERAL SYSTEM PARAMETERS
      READ(5,PARLST)
C     READ BRANCH + BOUND PARAMETERS
      READ(5,600)NPTHS,LBXX,NREJX,NMIN
C
C     READ STREAM,EQUIP + STREAM PATH INFORMATION
      READ(8,600)NNPR,NNSER,NNSM
      DO 625 JCH=1,2
      NNN=NNPR(JCH)
      DO 610 J=1,NNN
  610 READ(8,612)(SMCHX(II,J,JCH),II=1,NOCOMP)
      NNN=NNSM(JCH)
      DO 620 J=1,NNN
      READ(8,622)(SMCHA(II,J,JCH),II=1,8)
  620 READ(8,624)(SMCHB(II,J,JCH),II=1,7)
  625 CONTINUE
      DO 645 JCH=1,2
      JP1=IDJ(1,JCH)+1
      JP2=IDJ(NNPR(JCH),JCH)+NPTHS
      NP1=IDN(1,JCH)
      NP2=IDN(NNPR(JCH),JCH)
      READ(8,600)(NPATH(II),II=NP1,NP2)
      DO 640 JP=JP1,JP2
  640 READ(8,600)(JPATH(II,JP),II=1,8)
  645 CONTINUE
C
      READ(8,600)NEMCH
      DO 650 K=1,NEMCH
  650 READ(8,652)(EMCH(II,K),II=1,15)
C
C
C     READ IN INITIAL REFR LEVELS + COSTS
      WRITE(6,920)
```

```
      READ(5,600)NLL
      DO 904  I=1,NLL
      READ(5,902)X(I),Y(I)
  904 WRITE(6,902)X(I),Y(I)
C     SET UP ENERGY VALUE SPLINE
C     X=TEMP  ,  Y=VALUE
      NH=2
      NC=NLL
      X(NC+1)=TWAT
      Y(NC+1)=CWAT/(18.*DTW)
      X(NC+3)=TS
      Y(NC+3)=CS/(HVS*18.)
      X(NC+2)=0.5*(TWAT+TS)
      Y(NC+2)=0.5*(Y(NC+1)+Y(NC+3))
      DO 910  J=1,2
  910 CALL  SPLINE(J)
C
      CALL  ENEC
      CALL  BRBND(LBXX,NREJX,NMIN)
      CALL  ENDS
C
   11 FORMAT(3(/5E14.5))
  100 FORMAT(8A10)
  101 FORMAT(8A10//)
  600 FORMAT(10I5)
  612 FORMAT(8F10.4)
  622 FORMAT(5F6.0,5X,3F8.1)
  624 FORMAT(3F8.1,5X,F8.1,F10.0,F7.3,F8.1)
  652 FORMAT(3(2F5.0,3X),F12.0/5F9.1,3F9.0)
  902 FORMAT(F10.0,F10.7)
  920 FORMAT(/* REF LEVELS + COSTS -*/)
      END
```

```
      SUBROUTINE ENERGY
C
C     ***** COMMON DECK *****
C
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR,DTF(2)
      COMMON/EQUIP/EQUIP(15)
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SPLINE/NH,NC,X(10),Y(10),PM(10)
      COMMON/PATH/JPATH(8,200),NPATH(20),NPTHS
      COMMON/PLOPT/NEPT,NEOPT(40),NCOST(10)
C**   BLANK COMMON
      COMMON NNPR(2),NNSER(2),NNSM(2),SMCHA(8,50,2),SMCHB(7,50,2),
     1 SMCHX(8,10,2)
      COMMON NEMCH,EMCH(15,100)
C     ***
C
      DIMENSION TEX(2)
      DIMENSION SORT(20,2),KTAG(20),JPSV(8,20)
      DIMENSION REFD(10,2),REFS(10,2),PSRV(5,2)
C
C     INDEX DISPLACEMEMEMENT FUNCS FOR JPATH + NPATH ARRAYS
      IDJ(JPR,JCH)=NPTHS*((JPR-1)+NNPR(1)*(JCH-1))
      IDN(JPR,JCH)=JPR+NNPR(1)*(JCH-1)
C
C     JPATH ARRAY - EACH COL REPRESENTS ONE STREAM PROC PATH
C        1. NO OF EQUIPS IN PATH
C        2.-6. EQUIP NOS
C        7. COST OF PATH
C        8. ACTIVE(0)/INACTIVE(1) FLAG
C
C*****
      ENTRY ENEC
C*****
C     COMPUTES EQUIPS INVOLVING REFR + SALE (TYPES 1+30)
C     SUMS + SORTS COSTS FOR ALL STREAM PROCESSING PATHS
C
C     SCAN EMCH MATRIX
      DO 50 KE=1,NEMCH
      IF(EMCH(2,KE).NE.30.) GO TO 10
      IOP=30
      GO TO 12
   10 IF(EMCH(4,KE).NE.203.) GO TO 50
      IOP=1
      IS2=3
C     LOAD INPUT STREAM INTO SIN(1)
   12 JS=EMCH(3,KE)
      JCH=(3-ISIGN(1,JS))/2
      JN=IABS(JS)
      JPR=SMCHA(1,JN,JCH)
      CALL MVFSM(1,JN,JCH,JPR)
C     LOAD EQUIP WORKING VECTOR
      DO 14 K=1,15
   14 EQUIP(K)=EMCH(K,KE)
      IF(IOP.EQ.30) GO TO 30
C
C**   REF - FIND TEMP LEVEL LE TEX(2)
      TEX(1)=EQUIP(10)
      AP=APRR
      IF(TEX(1).LT.TRRR) AP=ARRR
      T2=TEX(1)-AP
      NCC=NC+1
      DO 20 L=1,NCC
   20 IF(X(L).GT.T2) GO TO 22
   22 LL=L-1
      TEX(2)=X(LL)
C     COMPUTE EXCHANGER + SUM COSTS
      CALL HXER(IOP,IS2,TEX,Q)
      EQUIP(10)=TEX(1)
      EQUIP(14)=Y(LL)*HRS*ABS(Q)
      EQUIP(15)=EQUIP(15)+EQUIP(14)
      GO TO 40
C
```

```
C**    SALE - PUT TEMP DISPL=DTF*APRR
   30  TEX=EQUIP(9)
       IF(TEX.LE.0.) TEX=TWAT
       DT=DTF(2)*APRR
       CALL SVALUE(JCH,TEX,DT,VALUE)
C
C*     LOAD EQUIP 7-15 INTO EMCH
   40  DO 42 K=7,15
   42  EMCH(K,KE)=EQUIP(K)
       WRITE(6,44)(EMCH(K,KE),K=1,15)
   50  CONTINUE
C      ************************
C
C*     SUM COSTS FOR EACH STREAM PROCESSING PATH IN JPATH ARRAY
C      FOR EACH PROCESS/PROCESS MATCH CHARGE HALF TO EACH STREAM
C      KEEP SEPARATE TOTAL FOR COL 1 (PRE-PROC) COSTS
       WRITE(6,400)
       COSTPP=0.
C
       DO 80 JCH=1,2
       NNN=NNPR(JCH)
       DO 80 J=1,NNN
       IF(J.LE.NNSER(JCH)) GO TO 80
       NCJ=IDJ(J,JCH)
       NCN=IDN(J,JCH)
C
C      SUM 1ST (PRE-PROC) COL
       NCOL=NCJ+1
       NNR=JPATH(1,NCOL)+1
       COST1=0.
       IF(NNR.EQ.1) GO TO 62
       DO 60 K=2,NNR
       NEQ=JPATH(K,NCOL)
   60  COST1=COST1+EMCH(15,NEQ)
       COSTPP=COSTPP+COST1
   62  JPATH(7,NCOL)=COST1
C
C      SUM OTHER COLS + SORT IN ORDER OF INCR COST
       NPS=NPATH(NCN)
       NCJ=NCJ+1
       DO 70 NC=1,NPS
       NCOL=NC+NCJ
       NNR=JPATH(1,NCOL)+1
       COST=COST1
       DO 66 K=2,NNR
       NEQ=JPATH(K,NCOL)
       CST=EMCH(15,NEQ)/2.
C      IS THIS A PROC/PROC MATCH - EXCL. PSEUDO-SERVICE
       DO 64 I=3,4
       J2=EMCH(I,NEQ)
   64  IF(J2.EQ.0.OR.J2.GT.200) CST=CST*2.
   66  COST=COST+CST
       SORT(NC,1)=JPATH(7,NCOL)=COST
C      SAVE JPATH COLUMN
       DO 68 NR=1,7
   68  JPSV(NR,NC)=JPATH(NR,NCOL)
   70  CONTINUE
C
C      SORT + REPLACE IN ORDER
       CALL TGSORT(SORT,KTAG,NPS,-1)
       WRITE(6,401)J
       DO 74 NC=1,NPS
       NCOL=NC+NCJ
       NCC=KTAG(NC)
       DO 72 NR=1,7
   72  JPATH(NR,NCOL)=JPSV(NR,NCC)
   74  WRITE(6,402)NCOL,(JPATH(II,NCOL),II=1,7)
   80  CONTINUE
       RETURN
C
C*****
       ENTRY ENDS
C*****
C      WRITES LIST OF EQUIPMENT VECTORS FOR OPTIMUM PLANT
```

```
C     COMPUTES LISTS OF REF DEMANDS , SALES + PSEUDO SERVICE USAGES
C     ALSO COMPUTES COST SUB-TOTALS
C
      WRITE(6,130)
      JCS=JCD=NS=ND=NPS=0
C
      DO 100 KE=1,NEPT
      NE=NEOPT(KE)
      WRITE(6,132)(EMCH(K,NE),K=1,15)
C     DOES EQUIP INVOLVE REF DEMAND OR SALE OR PSEUDO-SERVICE
      IF(EMCH(4,NE).EQ.203.) GO TO 84
      IF(EMCH(3,NE).GT.250..OR.EMCH(4,NE).GT.250.) GO TO 90
      IF(EMCH(2,NE).EQ.30.) GO TO 86
      GO TO 100
C
C*    REF DEMAND
   84 ND=ND+1
      SORT(ND,1)=EMCH(10,NE)
      SORT(ND,2)=-EMCH(7,NE)
      JCD=JCD+IFIX(EMCH(14,NE))
      GO TO 100
C
C*    REF SALE
   86 NS=NS+1
      SORT(10+NS,1)=EMCH(7,NE)
      SORT(10+NS,2)=EMCH(3,NE)
      JCS=JCS+IFIX(EMCH(14,NE))
      GO TO 100
C
C*    PSEUDO-SERVICE USAGE
   90 DO 92 J=3,4
   92 IF(EMCH(J,NE).GT.250.) GO TO 94
   94 STR=EMCH(J,NE)-300.
      IF(NPS.EQ.0) GO TO 97
C     HAS STREAM BEEN ENTERED
      DO 96 N=1,NPS
   96 IF(STR.EQ.PSRV(N,2)) GO TO 98
   97 N=NPS=NPS+1
      PSRV(N,1)=0.
C     ENTER STREAM + USAGE
   98 PSRV(N,1)=PSRV(N,1)+EMCH(10,NE)
      PSRV(N,2)=STR
  100 CONTINUE
C
C     SORT REF ARRAYS IN ORDER OF INCR TEMP
      DO 105 I=1,2
      IF(I.EQ.1) NN=ND
      IF(I.EQ.2) NN=NS
      II=10*(I-1)
      CALL TGSORT(SORT(II+1,1),KTAG,NN,-1)
      DO 105 J=1,NN
      JJ=KTAG(J)
      DO 105 K=1,2
      IF(I.EQ.1) REFD(J,K)=SORT(JJ,K)
  105 IF(I.EQ.2) REFS(J,K)=SORT(JJ+10,K)
C
C     SET COSTS + WRITE OUT DATA
      NCOST(2)=NCOST(1)-JCD-JCS
      NCOST(3)=COSTPP
      NCOST(4)=NCOST(2)-NCOST(3)
      NCOST(5)=JCD
      NCOST(6)=JCS
      WRITE(6,110)
      IF(ND.GT.0) WRITE(6,120)((REFD(N,I),I=1,2),N=1,ND)
      WRITE(6,122)
      IF(NS.GT.0) WRITE(6,120)((REFS(N,I),I=1,2),N=1,NS)
      WRITE(6,122)
      IF(NPS.GT.0) WRITE(6,120)((PSRV(N,I),I=1,2),N=1,NPS)
      WRITE(6,124)(NCOST(I),I=1,6)
      RETURN
C
   44 FORMAT(/* EQUIP*/3(2F5.0,3X),F12.0,5F9.1,3F9.0)
  110 FORMAT(///* REFD,REFS,PSRV ARRAYS -*/)
  120 FORMAT(2F10.0)
```

```
122 FORMAT(/)
124 FORMAT(//* TOTAL PLANT COST SUMMARY -*//* TOTAL*,39X,I10/* PROCESS
   1 NETWORK (EXCL. REFR. DEMANDS + SALES)*,I10/* PRE-PROCESSING (PRES
   2SURE SPECS)*,14X,I10/* POST-PROCESSING (TEMPERATURE SPECS)*,10X,I1
   30/* REFRIGERATION DEMANDS*,24X,I10/* STREAM SALES*,33X,I10)
130 FORMAT(////* EQUIPMENT DETAILS -*/)
132 FORMAT(3(2F5.0,3X),F12.0,5F9.1,3F9.0)
400 FORMAT(////* JPATH ARRAY (SORTED) -*)
401 FORMAT(/I3/)
402 FORMAT(I5,5X,I5,2X,5I5,I10)
    END
```

```
      SUBROUTINE BRBND(LBXX,NREJX,NMIN)
C
C     PERFORMS BRANCH + BOUND OPT ON STREAM PROC PATHS (JPATH ARRAY)
C     ROUTINE ALLOWS UP TO 3 LEVELS OF BRANCHING
C
      COMMON/PATH/JPATH(8,200),NPATH(20),NPTHS
      COMMON/PLOPT/NEPT,NEOPT(40),NCOST(10)
C$    BLANK COMMON
      COMMON NNPR(2),NNSER(2),NNSM(2),SMCHA(8,50,2),SMCHB(7,50,2),
     1 SMCHX(8,10,2)
      COMMON NEMCH,EMCH(15,100)
      COMMON COST(500),KTAG(500),NOPL(500)
C
      DIMENSION JSA(40),JCNT(40),JSGN(40),JSE(2,20)
      DIMENSION JSBP(2),NCRJ(40),JHIST(20)
      INTEGER BPP(3),BPR(3,20,3),NBP(3)
      DIMENSION LLJ(20),LLN(20),NPW(20),NPA(20),NOC(20)
C
C...  PARAMETERS -
C                   LBXX  - NO OF LEVELS OF BRANCHING
C                   NREJX - MIN NO OF REJECTIONS FOR PROBLEM
C                   NMIN  - MIN NO OF PROBLEMS AT ANY LEVEL
C
C     JPATH ARRAY - EACH COL REPRESENTS ONE STREAM PROC PATH
C         1. NO OF EQUIPS IN PATH
C         2.-6. EQUIP NOS
C         7. COST OF PATH
C         8. ACTIVE(0)/INACTIVE(GT 0) FLAG
C
C     INDEX DISPLACEMEMENT FUNCS FOR JPATH + NPATH ARRAYS
      IDJ(JPR,JCH)=NPTHS*((JPR-1)+NNPR(1)*(JCH-1))
      IDN(JPR,JCH)=JPR+NNPR(1)*(JCH-1)
C
      NPRH=NNPR(1)-NNSER(1)
      NPRC=NNPR(2)-NNSER(2)
      NPRR=NPRH+NPRC
C*    SET UP COL INDEX VECTORS FOR JPATH,NPATH (1ST PROC COL FOR JPATH)
      L=0
      DO 8 JCH=1,2
      NNN=NNPR(JCH)
      DO 8 J=1,NNN
C+++  NEGLECT PSEUDO-SERVICE STREAMS
      IF(J.LE.NNSER(JCH)) GO TO 8
      L=L+1
      LLJ(L)=IDJ(J,JCH)+1
      LLN(L)=IDN(J,JCH)
    8 CONTINUE
C
C     SET SHIFTING CONSTANTS FOR ENCODING PLANT NOS
      NOC(NPRR)=1
      DO 10 JS=2,NPRR
      JC=NPRR+2-JS
   10 NOC(JC-1)=NOC(JC)*NPTHS
C**********************************************************************
C
      JCSTX=NOPX=10000000
      NPRX=NPRX2=NPRX3=IACT=0
      ASSIGN 22 TO IZA
      GO TO 260
C
C**** COMPUTE ALL BOUNDING PROBLEMS *****
C
   22 DO 201 IPAS=1,2
      LBX=LBXX
      IF(IPAS.EQ.1) LBX=1
      IF(IPAS.EQ.2)WRITE(6,436)JCSTX,NOPX
C
C**   SET UP 1ST LEVEL OF BRANCHING **
      LB=1
      IF(IPAS.EQ.2) GO TO 25
      ASSIGN 24 TO IBPR
      GO TO 300
   24 NPRX1=NBPR+1
   25 DO 200 NPR1=1,NPRX1
C     ELIMINATE COLS INCOMP WITH NPR1
```

```
      NBP(1)=NPR=NPR1
      CALL ZEROI(NBP(2),2)
      NPRX=NPRX1
      LB=1
      ASSIGN 26 TO IBRJ
      GO TO 340
   26 IF(LBX.EQ.1) GO TO 32
C
C**   SET UP 2ND LEVEL OF BRANCHING **
      LB=2
      ASSIGN 30 TO IBPR
      GO TO 300
   30 NPRX2=NBPR+1
   32 DO 190 NPR2=1,NPRX2
      IF(LBX.EQ.1) GO TO 42
      NBP(2)=NPR=NPR2
      NBP(3)=0
      NPRX=NPRX2
      LB=2
      ASSIGN 34 TO IBRJ
      GO TO 340
   34 IF(LBX.EQ.2) GO TO 42
C
C**   SET UP 3RD LEVEL OF BRANCHING **
      LB=3
      ASSIGN 40 TO IBPR
      GO TO 300
   40 NPRX3=NBPR+1
   42 DO 180 NPR3=1,NPRX3
      IF(LBX.LE.2.OR.NPRX3.EQ.1) GO TO 50
      NBP(3)=NPR=NPR3
      NPRX=NPRX3
      LB=3
      ASSIGN 50 TO IBRJ
      GO TO 340
   50 IF(IPAS.EQ.2) GO TO 110
C*****************************
C
C***  COMPUTE CLOSER UPPER BOUND ****
C     COMPUTE BOUNDS FOR ALL STREAMS (IPAS=1 ONLY - PRIM LEVEL PROBLEMS)
C
   80 IF(NPR1.EQ.NPRX1) GO TO 200
      KPAS=2
      DO 100 LS=1,NPRR
      CALL ZEROI(NPW,NPRR)
      KSE=NSE=ISE=NSA=0
      JS=LS
C     ENTER LOWEST COST COMPAT PATH FOR JS
   82 ASSIGN 84 TO IPAR
      GO TO 250
   84 DO 86 NCOL=NC1,NC2
      IF(JPATH(8,NCOL).GT.0) GO TO 86
      IF(KSE.EQ.0) GO TO 83
C     SEARCH FOR EQUIP NSEQ
      NNR=JPATH(1,NCOL)+1
      DO 95 NR=2,NRR
   95 IF(JPATH(NR,NCOL).EQ.NSEQ) GO TO 83
      GO TO 86
C     TEST PATH NCOL + ENTER IF COMPAT WITH JSA (ICOMP=1)
   83 ASSIGN 85 TO JSENT
      GO TO 230
   85 IF(ICOMP.EQ.1) GO TO 87
   86 CONTINUE
      GO TO 100
   87 NPW(JS)=NCOL-NCJ
C
C     ENTER NEXT OP SIGN INPUT , OTHERWISE MOVE TO NEXT JS
   88 KSE=0
      IF(JSE(1,ISE+1).NE.0) GO TO 92
      DO 90 JS=1,NPRR
   90 IF(NPW(JS).EQ.0) GO TO 82
      GO TO 96
   92 KSE=1
      ISE=ISE+1
```

```
      NSEQ=JSE(1,ISE)
      JSS=JSE(2,ISE)
C     IDENTIFY JS FOR JSS + ENTER JS - IF NOT ALREADY ENTERED
      ASSIGN 94 TO IJS
      GO TO 255
   94 IF(NPW(JS).GT.0) GO TO 88
      GO TO 82
C
C     CHECK ENTRY COUNT VECTOR + COST - IF OK REPLACE BOUND
   96 DO 97 N=1,NSA
   97 IF(JCNT(N).NE.2) GO TO 100
      ASSIGN 98 TO INOP
      GO TO 210
   98 WRITE(6,517)ICST,NOP
      IF(ICST.GE.JCSTX) GO TO 100
      NOPX=NOP
      JCSTX=ICST
      WRITE(6,412)JCSTX,NOPX,LS
  100 CONTINUE
      GO TO 192
C****************************
C
C***  USE BOUND TO REJECT ALL PATHS LEADING TO PLANTS WITH COST GE BOUND
C
  110 NREJ=0
  120 DO 130 JS=1,NPRR
      ASSIGN 121 TO IPAR
      GO TO 250
  121 DO 126 NCOL=NC1,NC2
      IF(JPATH(8,NCOL).GT.0) GO TO 126
      NPW(JS)=NCOL-NCJ
C
C     SCAN ALL STREAMS NE JS + ADD IN LOWEST COST ACT PATH
      DO 124 LS=1,NPRR
      IF(LS.EQ.JS) GO TO 124
      NCL=NCCL=LLJ(LS)
  122 NCCL=NCCL+1
      IF(JPATH(8,NCCL).GT.0) GO TO 122
      NPW(LS)=NCCL-NCL
  124 CONTINUE
C     COMPUTE COST + IF GE BOUND SET INACT FLAG (TO LBX)
      ASSIGN 125 TO INOP
      GO TO 210
  125 IF(ICST.LT.JCSTX) GO TO 126
      JPATH(8,NCOL)=LBX
      NPA(JS)=NPA(JS)-1
      IF(NPA(JS).EQ.0) GO TO 175
      NREJ=NREJ+1
      NCRJ(NREJ)=NCOL
  126 CONTINUE
  130 CONTINUE
      IF(NREJ.GT.0) WRITE(6,414)(NCRJ(II),II=1,NREJ)
C****************************
C
C***  MAKE UP ALL POSSIBLE PLANTS FROM ACTIVE PATHS ****
C
      ISZ=1
      DO 132 J=1,NPRR
  132 ISZ=ISZ*NPA(J)
      WRITE(6,434)(NPA(II),II=1,NPRR),ISZ
C     INITIALIZE NPW - TO FIRST ACTIVE PLANT
      DO 134 II=1,NPRR
      NCOL=NCJ=LLJ(II)
  133 NCOL=NCOL+1
      IF(JPATH(8,NCOL).GT.0) GO TO 133
  134 NPW(II)=NCOL-NCJ
      NN=NNSER(1)+1
      NPT1=NPATH(NN)
      NPL=0
      GO TO 142
C
C     COMPUTE NPW VECTOR FOR NEXT ACT PLANT
  136 DO 140 LS=1,NPRR
      JS=NPRR-LS+1
```

```
      ASSIGN 137 TO IPAR
      GO TO 250
  137 IEX=0
C     INCREMENT NPW ELEMENT
  138 NPW(JS)=NPW(JS)+1
C     HAVE ALL COMBINATIONS BEEN COVERED
      IF(NPW(1).GT.NPT1) GO TO 150
C     IF NPATH(NCN) EXCEEDED , RE-SET TO 1
      IF(NPW(JS).GT.NPS) NPW(JS)=IEX=1
      NCOL=NCJ+NPW(JS)
      IF(JPATH(8,NCOL).GT.0) GO TO 138
  140 IF(IEX.EQ.0) GO TO 142
C
C     COMPUTE PLANT NO + COST + ENTER IN SORT VECTORS - IF COST LT BOUND
  142 ASSIGN 144 TO INOP
      GO TO 210
  144 IF(ICST.GE.JCSTX) GO TO 136
      NPL=NPL+1
      COST(NPL)=ICST
      NOPL(NPL)=NOP
      GO TO 136
C
  150 WRITE(6,418)NPL
C****************************
C
C**   SORT   **
      CALL TGSORT(COST,KTAG,NPL,-1)
C
C*    FIRST (FEASIBLE) ENTRY IN TAG LIST IS OPT PLANT - REPLACE BOUND
C
      KPAS=1
      DO 160 L=1,NPL
C     RE-CONSTRUCT NPW VECTOR FROM PLANT NO
      LL=KTAG(L)
      ICST=COST(LL)
      NOP=NOPL(LL)
      ASSIGN 154 TO INPW
      GO TO 220
  154 NSA=0
C     CHECK NPW FOR FEASIBILITY - FOR EACH COL ICOMP=1 INDICATES FEAS
      DO 156 JS=1,NPRR
      NCOL=NPW(JS)+LLJ(JS)
      ASSIGN 156 TO JSENT
      GO TO 230
  156 IF(ICOMP.EQ.0) GO TO 160
C     CHECK ENTRY COUNT VECTOR - IF OK REPLACE BOUND
      DO 158 N=1,NSA
  158 IF(JCNT(N).NE.2) GO TO 160
      JCSTX=ICST
      NOPX=NOP
      WRITE(6,412)JCSTX,NOPX,L
      GO TO 176
  160 CONTINUE
C
      WRITE(6,422)
      GO TO 176
  175 WRITE(6,425)NCOL
C
  176 GO TO (192,182,178),LBX
C     RE-SET ACT FLAGS  GE  IACT
  178 IACT=3
      ASSIGN 180 TO IZA
      GO TO 260
  180 CONTINUE
  182 IACT=2
      ASSIGN 190 TO IZA
      GO TO 260
  190 CONTINUE
  192 IACT=1
      ASSIGN 200 TO IZA
      GO TO 260
  200 CONTINUE
  201 CONTINUE
C*************************************************************
```

```
C
C**   COMPUTE + SORT EQUIP NO VECTOR FOR OPT PLANT (INCL PRE-PROC EQUIP) *
C
      NCOST(1)=JCSTX
      NOP=NOPX
      ASSIGN 202 TO INPW
      GO TO 220
  202 NEPT=0
      DO 208 JS=1,NPRR
      NCOL=LLJ(JS)
      DO 208 M=1,2
      IF(M.EQ.2) NCOL=NCOL+NPW(JS)
      NNR=JPATH(1,NCOL)+1
      IF(NNR.EQ.1) GO TO 208
      DO 206 NR=2,NNR
      NEQ=JPATH(NR,NCOL)
      IF(NEPT.EQ.0) GO TO 205
C     HAS NEQ BEEN ENTERED
      DO 204 N=1,NEPT
  204 IF(NEQ.EQ.IFIX(COST(N))) GO TO 206
  205 NEPT=NEPT+1
      COST(NEPT)=NEQ
  206 CONTINUE
  208 CONTINUE
C
C     SORT INTO INCR ORDER
      CALL TGSORT(COST,KTAG,NEPT,-1)
      DO 209 J=1,NEPT
      JJ=KTAG(J)
  209 NEOPT(J)=COST(JJ)
      WRITE(6,420)NOPX,JCSTX,(NEOPT(II),II=1,NEPT)
      RETURN
C
C
C*****************************************************************
C**   ROUTINE TO COMPUTE NOP + ICST (PLANT NO + COST) FROM NPW VECTOR **
  210 NOP=ICST=0
      DO 212 KK=1,NPRR
      KCOL=NPW(KK)+LLJ(KK)
      NOP=NOP+NPW(KK)*NOC(KK)
  212 ICST=ICST+JPATH(7,KCOL)
      GO TO INOP,(98,125,144)
C*****
C**   ROUTINE TO COMPUTE NPW VECTOR FROM NOP **
  220 NOPP=NOP
      DO 222 KK=1,NPRR
      NPW(KK)=(NOPP/NOC(KK))
  222 NOPP=NOPP-(NPW(KK)*NOC(KK))
      GO TO INPW,(154,202)
C*****
C**   ROUTINE TO ENTER COL NCOL (STREAM JS) INTO PLANT + CHECK COMPAT **
C     IF COMPATIBLE ICOMP IS SET TO 1
C     INPUTS ARE ENTERED IN JSA VECTOR + ENTRIES COUNTED IN JCNT VECTOR
C     COUNT IX=1 FOR PROC/PROC MATCH , 2 FOR PROC/SERV MATCH V/R OR SALE
C     SIGN OF  JS  FOR ENTRY IS STORED IN JSGN VECTOR
C     TEST ON NPAS=1 , ENTER ON NPAS=2
C
  230 ICOMP=0
      IF(NSA.EQ.0) CALL ZEROI(JSA,160)
      NHC=2*(NPRH-JS)+1
      NRR=JPATH(1,NCOL)+1
C
      DO 240 NPAS=1,KPAS
      DO 240 NR=2,NRR
      NEQ=JPATH(NR,NCOL)
      IX=1
      DO 242 I=3,4
      NS=EMCH(I,NEQ)
  242 IF(NS.EQ.0.OR.NS.GT.200) IX=2
      DO 239 NI=3,4
      NS=EMCH(NI,NEQ)
      IF(NS.EQ.0.OR.NS.GT.200) GO TO 239
C     HAS STREAM BEEN ENTERED
      IPREV=1
```

```
      DO 232 N=1,NSA
  232 IF(NS.EQ.JSA(N)) GO TO 233
      IPREV=0
  233 IF(NPAS.EQ.2) GO TO 237
C
C*    CHECKING ROUTINE
      IF(IPREV.EQ.0) GO TO 234
C     CHECK JSGN - IF SAME SIGN AS NHC , REJECT
      IF((JSGN(N)*NHC).GT.0) GO TO 246
C     CHECK ENTRY COUNTER - IF GT 2 , REJECT
      IF((JCNT(N)+IX).GT.2) GO TO 246
      IF(IPREV.EQ.1) GO TO 236
C     IF NS IS NEW STREAM , CHECK HISTORY
C     - IF STREAM FOUND WITH ENTRY COUNTER EQ 2 , REJECT
  234 CALL SHIST(NS,NHIST,JHIST)
      DO 235 NJ=1,NSA
      DO 235 NH=1,NHIST
  235 IF(JSA(NJ).EQ.JHIST(NH).AND.JCNT(NJ).EQ.2) GO TO 246
  236 IF(KPAS.EQ.2) GO TO 239
C
C*    ENTERING ROUTINE
  237 IF(IPREV.EQ.1) GO TO 238
      N=NSA=NSA+1
      JSA(N)=NS
      JSGN(N)=NHC
  238 JCNT(N)=JCNT(N)+IX
C     FOR NPAS EQ 2 SAVE EQUIP + STREAM NO FOR INPUT OF OP SIGN FROM JS
      IF(NPAS.EQ.1.OR.(NS*NHC).GT.0) GO TO 239
      NSE=NSE+1
      JSE(1,NSE)=NEQ
      JSE(2,NSE)=NS
  239 CONTINUE
  240 CONTINUE
      ICOMP=1
  246 GO TO JSENT,(85,156)
C*****
C**   ROUTINE TO SET UP PARAMS FOR PATH SCAN FOR STREAM JS **
  250 NCJ=LLJ(JS)
      NCN=LLN(JS)
      NPS=NPATH(NCN)
      NC1=NCJ+1
      NC2=NCJ+NPS
      GO TO IPAR,(84,121,137,264,302,342)
C*****
C**   ROUTINE TO SUPPLY  JS  FOR STREAM JSS **
C     ALSO SUPPLIES SEC STREAM NO + ACIVE/INACTIVE FLAG
  255 JCH=(3-ISIGN(1,JSS))/2
      JST=IABS(JSS)
      JPR=SMCHA(1,JST,JCH)
      JSEC=SMCHA(2,JST,JCH)
      JACT=SMCHA(3,JST,JCH)
      IF(JCH.EQ.1) JS=JPR-NNSER(1)
      IF(JCH.EQ.2) JS=JPR+NPRH-NNSER(2)
      GO TO IJS,(94,305,308,366)
C*****
C**   ROUTINE TO ZERO ACT FLAGS PRES  GE  IACT + RE-SET NPA VECTOR **
  260 DO 266 KK=1,NPRR
C     RE-SET NPA TO NPATH
      JS=KK
      ASSIGN 264 TO IPAR
      GO TO 250
  264 NPA(KK)=NPS
      DO 266 KCOL=NC1,NC2
      IF(JPATH(8,KCOL).GE.IACT) JPATH(8,KCOL)=0
  266 IF(JPATH(8,KCOL).GT.0) NPA(KK)=NPA(KK)-1
      GO TO IZA,(22,180,190,200,316)
C*****
C
C***  IDENTIFY P/P MATCHES FOR BOUNDING PROBS (PLACE IN BPR ARRAY) ****
C     VALID BRANCHING PROBLEMS AT LEVEL (LB) ...
C     1. PRIM/PRIM (INCL V/R EX)  - FIRST EQUIP ROW OF JPATH
C     2. SEC/(PRIM..) (EXCL V/R)  - SECOND EQUIP ROW OF JPATH
C     3. AS FOR 2. + TERT/(PRIM..)  - THIRD EQUIP ROW OF JPATH
C     FOR NBPR LT NMIN , ADD PROBLEMS FROM 1. UNTIL NBPR=NMIN
```

```
C       DISCARD PROBLEMS WHICH PRODUCE LT NREJX REJECTIONS
C
  300 NBPR=JPAS=0
      LR1=LR2=LB+1
      IF(LB.EQ.3) LR1=LR1-1
  301 DO 322 MS=1,NPRR
      JS=MS
      ASSIGN 302 TO IPAR
      GO TO 250
  302 MC1=NC1
      MC2=NC2
      DO 320 MCOL=MC1,MC2
      IF(JPATH(8,MCOL).GT.0) GO TO 320
      IVR=0
      DO 318 LBB=LR1,LR2
  303 BPP(1)=NEQ=JPATH(LBB,MCOL)
      IF(NEQ.EQ.0) GO TO 320
      IF(IVR.EQ.1) GO TO 307
C
C       IF EITHER INLET STREAM INACTIVE , SKIP COL
C+      (INACTIVE INLET - 2ND STAGE V/R COMPR , OR A/C INLET)
C       SKIP ALSO IF SEC STREAM NO FOR EITHER IS INVALID FOR LEVEL LB
      DO 306 I=3,4
      BPP(I-1)=JSS=EMCH(I,NEQ)
C       REJECT IF NOT PROC/PROC MATCH OR V/R COMPR (TYPE 11.)
      IF(JSS.GT.200) GO TO 318
      IF(JSS.NE.0) GO TO 304
      IF(EMCH(2,NEQ).EQ.11.) GO TO 306
      GO TO 318
  304 ASSIGN 305 TO IJS
      GO TO 255
  305 IF(JACT.EQ.1.OR.JSEC.GE.LB) GO TO 320
  306 CONTINUE
C
C       IF OUTLET STREAM INACTIVE , SKIP TO NEXT ROW
C+      (INACTIVE OUTLET - V/R COMPR OUTLET)
  307 DO 309 I=5,6
      JSS=EMCH(I,NEQ)
      IF(JSS.EQ.0) GO TO 309
      ASSIGN 308 TO IJS
      GO TO 255
  308 IF(JACT.EQ.1) GO TO 310
  309 CONTINUE
      GO TO 311
C
  310 IVR=1
      LBB=LBB+1
      GO TO 303
C
  311 IF(IVR.EQ.1) BPP(3)=EMCH(4,NEQ)
      IF(LB.EQ.1) GO TO 313
C       IS NEQ SAME AS BPR FOR HIGHER LEVEL
      LLL=LB-1
      DO 312 LL=1,LLL
      KPR=NBP(LL)
  312 IF(NEQ.EQ.BPR(1,KPR,LL)) GO TO 318
C       ENTER MATCH IN BPR - IF NOT ALREADY ENTERED
  313 DO 314 NN=1,NBPR
  314 IF(BPR(1,NN,LB).EQ.NEQ) GO TO 318
      NPR=NBPR=NBPR+1
      DO 317 I=1,3
  317 BPR(I,NPR,LB)=BPP(I)
C       CHECK NO OF REJS PRODUCED BY PROB - IF LT NREJX , REJECT PROB
      NRJ=1
      ASSIGN 315 TO IBRJ
      GO TO 376
  315 IF(NREJ.LT.NREJX) NBPR=NBPR-1
C       RE-SET NPA VECTOR
      IACT=LB
      ASSIGN 316 TO IZA
      GO TO 260
  316 IF(JPAS.EQ.1.AND.NBPR.GE.NMIN) GO TO 324
  318 CONTINUE
  320 CONTINUE
```

```
 322 CONTINUE
C
C     FOR NBPR LT NMIN , RE-SCAN FOR 1. PROBLEMS
      IF(NBPR.GE.NMIN.OR.JPAS.EQ.1) GO TO 324
      JPAS=1
      LR1=LR2=2
      GO TO 301
 324  WRITE(6,430)LB
      DO 326 I=1,3
 326  WRITE(6,405)(BPR(I,II,LB),II=1,NBPR)
      GO TO IBPR,(24,30,40)
C*****
C
C***  SCAN PATHS FOR INFEAS DUE TO MULT STREAM USE ****
C     (STREAMS IN BPR BUT NOT IN KEQ MATCH) - SET INACT FLAG
C**   MUST ALSO CHECK THAT BPR IS PRESENT
C
 340  WRITE(6,408)(NBP(LL),LL=1,LBX)
      NRJ=0
      IF(NPR.EQ.NPRX) GO TO 372
C     SET UP BPR + IDENTIFY JPATH SECTION(S) CONTAINING BPR
 376  KEQ=BPR(1,NPR,LB)
      DO 370 J=1,2
      JSS=JSA(J)=BPR(J+1,NPR,LB)
      ASSIGN 366 TO IJS
      GO TO 255
 366  JSBP(J)=JS
 370  CONTINUE
 372  NSA1=1
      NSA3=2
      NREJ=0
 341  NSA2=NSA3
C
      DO 362 JS=1,NPRR
      ASSIGN 342 TO IPAR
      GO TO 250
 342  DO 360 NCOL=NC1,NC2
      IF(JPATH(8,NCOL).GT.0) GO TO 360
C     SET BPR FLAG TO 1 IF BPR MUST BE FOUND IN NCOL
      KBPR=0
      IF(NPR.EQ.NPRX) GO TO 374
      IF(JS.EQ.JSBP(1).OR.JS.EQ.JSBP(2)) KBPR=1
 374  IVR=0
      NRR=JPATH(1,NCOL)+1
      DO 352 NR=2,NRR
      NEQ=JPATH(NR,NCOL)
      IF(NPR.NE.NPRX) GO TO 346
C*    EXTRA PROBLEM - REJECT IF ANY BOUNDING EQUIPS ARE FOUND
      NN=NPRX-1
      DO 344 N=1,NN
 344  IF(NEQ.EQ.BPR(1,N,LB)) GO TO 354
      GO TO 352
 346  IF(NEQ.EQ.KEQ) GO TO 360
      IF(IVR.EQ.1) GO TO 352
C     TEST STREAMS
      DO 350 NI=3,4
      NS=EMCH(NI,NEQ)
      DO 348 N=NSA1,NSA2
 348  IF(NS.EQ.JSA(N)) GO TO 351
 350  CONTINUE
      GO TO 352
 351  IF(EMCH(2,NEQ).NE.11.) GO TO 354
      IVR=1
 352  CONTINUE
C     IF BPR FLAG OR V/R FLAG SET - REJECT
      IF(KBPR.NE.0.OR.IVR.NE.0) GO TO 354
      GO TO 360
C
C+    SET INACT FLAG (TO LB) + ADD (ILLEGAL) RESIDUALS TO JSA
 354  JPATH(8,NCOL)=LB
      NPA(JS)=NPA(JS)-1
      IF(NPA(JS).EQ.0) GO TO 175
      NREJ=NREJ+1
      NCRJ(NREJ)=NCOL
```

```
      DO 358 NO=5,6
      NS=EMCH(NO,NEQ)
      IF(NS.EQ.0) GO TO 358
C     HAS NS BEEN ENTERED
      DO 356 N=NSA2,NSA3
  356 IF(NS.EQ.JSA(N)) GO TO 358
      NSA3=NSA3+1
      JSA(NSA3)=NS
  358 CONTINUE
  360 CONTINUE
  362 CONTINUE
      IF(NSA3.EQ.NSA2) GO TO 364
C     RE-SCAN FOR RESIDUALS
      NSA1=NSA2+1
      GO TO 341
C
  364 IF(NRJ.EQ.0.AND.NREJ.GT.0) WRITE(6,432)(NCRJ(II),II=1,NREJ)
      GO TO IBRJ,(26,34,50,315)
C
  405 FORMAT(25I5)
  408 FORMAT(/* ....PROBLEM NO*,3I4)
  412 FORMAT(/4H ***,* IMPROVED BOUND , PLANT NO. - *,I10,5X,I14,I5/)
  414 FORMAT(* REJS DUE TO BOUND - *,25I4)
  418 FORMAT(* LIST SIZE*,I5)
  420 FORMAT(///* OPT PLANT - NO,COST*,5X,I14,I10//* EQUIP NOS -*/30I4)
  422 FORMAT(* NO FEASIBLE SOLUTIONS*)
  425 FORMAT(* PROBLEM SIZE ZERO  (NCOL =*,I3,*)*)
  430 FORMAT(/* NEW PROBLEM ARRAY AT LEVEL*,I4,* -*)
  432 FORMAT(* REJS DUE TO BPR - *,25I4)
  434 FORMAT(/* NPA VECTOR , TOT PROB SIZE -*,8I3,5X,*...*,I6)
  436 FORMAT(//* BEST BOUND FROM PASS 1 -*,I10,5X,I14//)
  517 FORMAT(I10,5X,I14)
      END
```

D REFRIGERATION UNIT

MAINR reads the input data for this section. The data consists mainly of refrigeration demand and purchased stream information. The routine sets up the energy cost spline based on previous refrigeration cost data and places the purchased streams in their correct refrigerant circuit array for subsequent use.

RUNIT is the routine which automatically generates a standard cascade refrigeration unit for a given set of refrigeration demands (temperature levels with associated cooling loads). The calculational procedure has been described in section 5.4 and is given here in graphical form in Figure II.3. The following points should be noted.

i) Refrigerant circuits are computed in increasing order of temperature, i.e. methane, ethylene, then propane. This is required for the correct direction of information transfer (condensation loads and purchased stream residuals) between circuits.

ii) The refrigeration unit serves as an energy cost updating routine. Cost data from previous computations are supplied to the routine in order to cost purchased stream energy. After computation of the refrigeration unit the routine combines these purchase costs with equipment capital and operating costs to produce a new set of energy cost figures for the next computation cycle.

STMOVR is the version of the stream handling utility routine for this section.

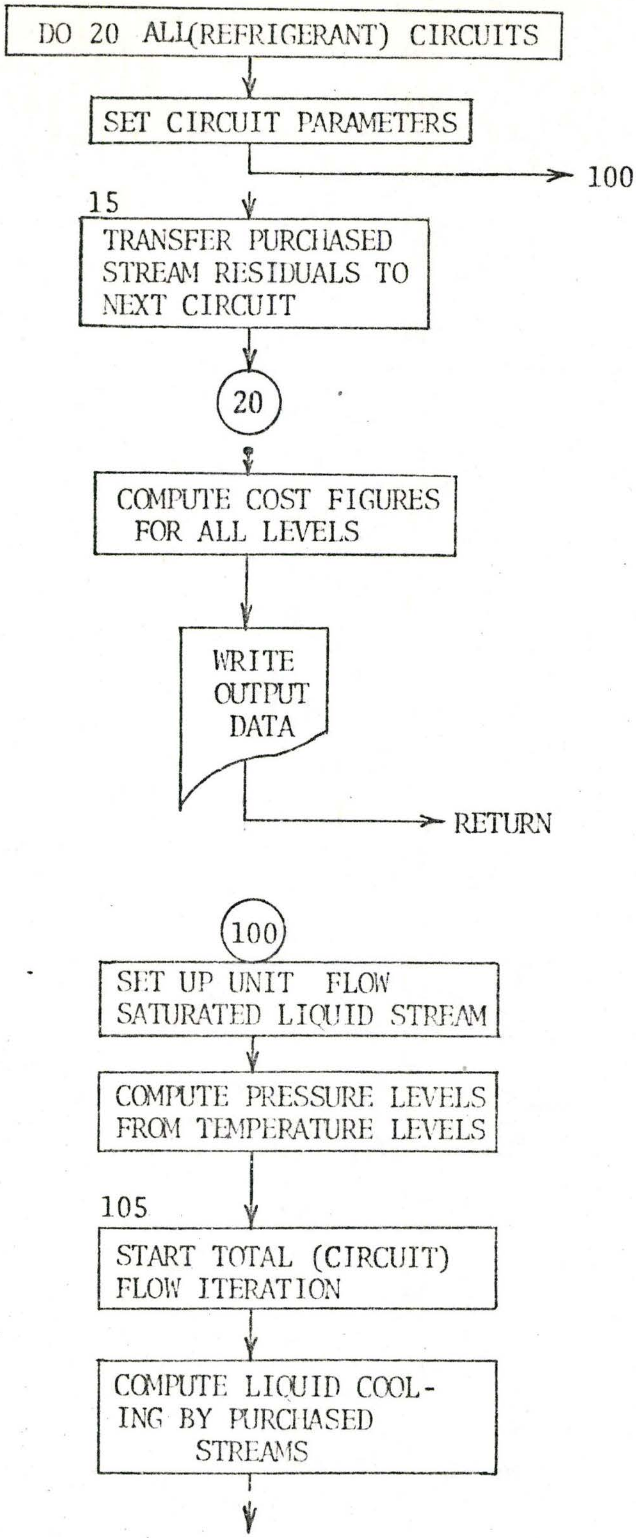


Figure II.3 RUNIT Algorithm (Continued on page 166).

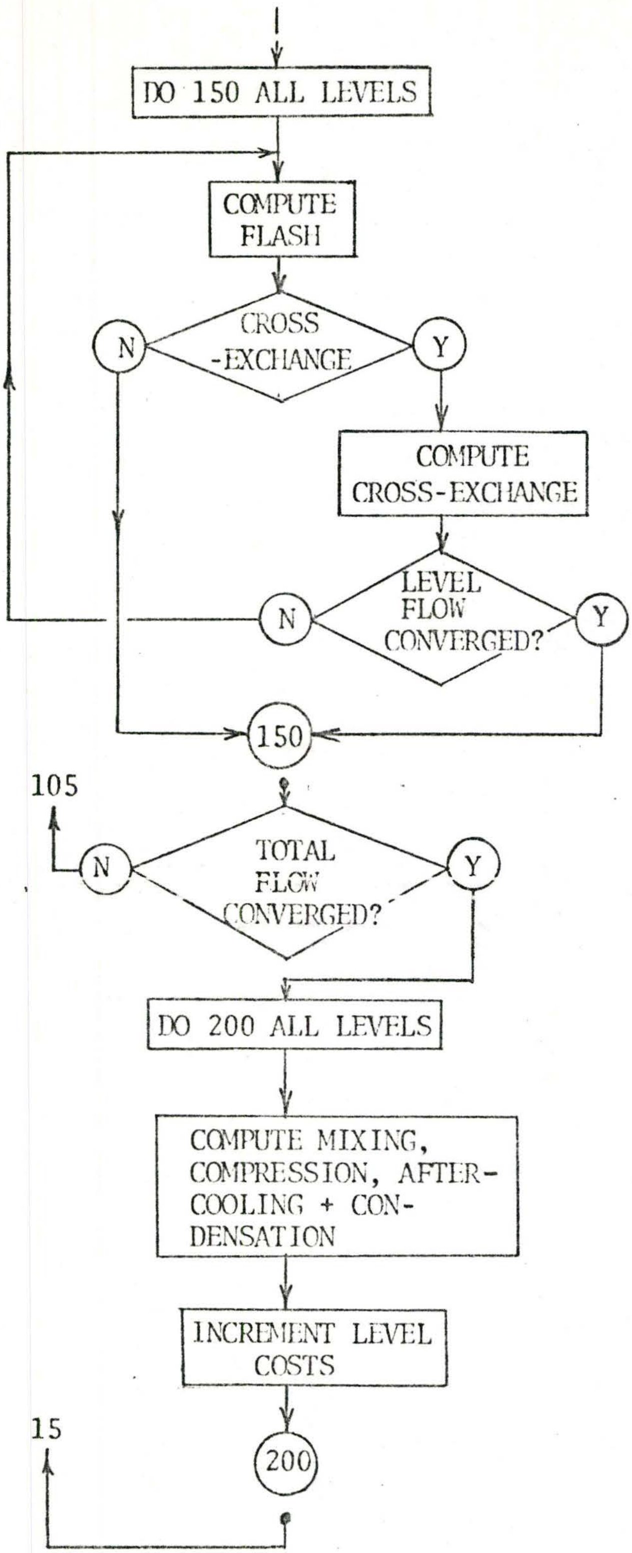


Figure II.3 RUNIT Algorithm (Continued).

```
      PROGRAM MAINR(INPUT=1001,OUTPUT=1001,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     *** COMMON DECK ***
C
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/PROP/COMPNT(8),APC(8),ATC(8),AVC(8),AMW(8),AOMEG(8),
     1 ADEL(8),AVW(8),APH(8),BET(8),GAM(8),DTA(8),BASEA(8),BASEB(8),
     2 ZCD(8),ALD(8)
      COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR,DTF(2)
      COMMON/SPLINE/NH,NC,X(10),Y(10),PM(10)
      COMMON/REFD/LRF(3),NLL,RLEV(10,2),TMSERV
C$    BLANK COMMON
      COMMON NSMRB(3),SMRB(7,20,3),SMRX(8,4,3)
C     ***
C
      INTEGER COMPNT
      DIMENSION TITLE(8),PROP(8,15)
      EQUIVALENCE (PROP,APC)
      NAMELIST/PARLST/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR,DTF
      NAMELIST/COMP/NOCOMP,COMPNT
C
C     READ COMPONENT INFORMATION
      READ(5,COMP)
C     READ COMPONENT PHYSICAL CONSTANTS
      DO 10 I=1,NOCOMP
   10 READ(5,11)(PROP(I,K),K=1,15)
C
C     READ TITLE
      READ(5,100)TITLE
      IF(EOF,5)1,2
    1 CALL EXIT
    2 WRITE(6,101)TITLE
C     READ GENERAL SYSTEM PARAMETERS
      READ(5,PARLST)
C
C     READ IN PRESENT REF LEVELS + COSTS
      READ(5,700)NLL
      DO 704 I=1,NLL
  704 READ(5,702)X(I),Y(I)
C     SET UP ENERGY VALUE SPLINE - X=TEMP , Y=VALUE
      NH=2
      NC=NLL
      X(NC+1)=TWAT
      Y(NC+1)=CWAT/(18.*DTW)
      X(NC+3)=TS
      Y(NC+3)=CS/(HVS*18.)
      X(NC+2)=0.5*(TWAT+TS)
      Y(NC+2)=0.5*(Y(NC+1)+Y(NC+3))
      DO 710 J=1,2
  710 CALL SPLINE(J)
C
C     READ IN NEW REF LEVELS
      READ(5,700)LRF,NLL
      DO 720 L=1,NLL
  720 READ(5,702)(RLEV(L,II),II=1,2)
C     READ IN SOLD STREAM INFORMATION + PLACE IN REFS ARRAYS
      CALL ZEROI(NSMRB,3)
      READ(5,700)NS
      DO 730 N=1,NS
      READ(5,700)IR
      NN=NSMRB(IR)=NSMRB(IR)+1
C     READ PROPS
      READ(5,702)(SMRB(II,NN,IR),II=1,7)
  730 READ(5,702)(SMRX(II,NN,IR),II=1,7)
      READ(5,702)TMSERV
C
      CALL RUNIT
C
   11 FORMAT(3(/5E14.5))
  100 FORMAT(8A10)
  101 FORMAT(8A10//)
  700 FORMAT(4I5)
  702 FORMAT(7F10.0)
      END
```

```
      SUBROUTINE RUNIT
C
C     COMPUTES THREE-TIER CASCADE REFRIGERATION SYSTEM
C     REFRIGERANTS - METHANE,ETHYLENE,PROPANE
C     UP TO 3 LEVELS OF EACH REFRIGERANT  (PURE REFRS ASSUMED)
C
C     SUBSCRIPT IR , 1 METHANE , 2 ETHYLENE , 3 PROPANE
C     RLEV - REFR DEMAND VECTORS  1. T , 2. Q
C     NLEV - NO OF LEVELS
C     EMR  - EQUIP MATRIX
C     SMRB - STREAM MATRIX (B SECTION)
C
C*****  COMMON DECK
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/EQUIP/EQUIP(15)
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
      COMMON/REFD/LRF(3),NLL,RLEV(10,2),TMSERV
      COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR,DTF(2)
C$    BLANK COMMON
      COMMON NSMRB(3),SMRB(7,20,3),SMRX(8,4,3)
      COMMON NEMR(3),EMR(15,15,3)
C*****
C
      DIMENSION TEX(2),DUM(8),DHV(3),SUMQ(3),COEF(4,3)
      DIMENSION TLEV(10),QLEV(10),PLEV(10),FLEV(10),FLOC(2),QRC(2)
      DIMENSION NCMP(3),TRFC(3),LLV(2,3)
      DIMENSION COST(10),CPS(3),CPE(3)
      DIMENSION NSP(3),NVP(3),NRSS(3)
      EQUIVALENCE (TLEV,RLEV(1,1)),(QLEV,RLEV(1,2))
      DATA COEF/-3.0424E2,6.3996E0,-4.2667E-2,9.3341E-5,
     1          -9.6343E2,1.0171E1,-3.6795E-2,4.5781E-5,
     2          -1.0155E3,8.0120E0,-2.1511E-2,1.9720E-5/
      DATA DHV/3000.,4500.,5500./
      DATA NCMP/2,3,6/
      DATA TRFC/305.,425.,545./
C
C     P  VS  T  FUNCTION FOR PURE REFS
      P(IR,T)=COEF(1,IR)+T*(COEF(2,IR)+T*(COEF(3,IR)+T*COEF(4,IR)))
C
C     ZERO MATRICES + MATRIX COUNTERS
      CALL ZEROI(NEMR,3)
      CALL ZERO(EMR,675)
      CALL ZERO(FLEV,10)
      CALL ZERO(COST,16)
      CALL ZERO(SUMQ,3)
C
C     SET PURCHASED STREAM COUNTER
C     PURCHASED STREAMS ARE ALREADY STORED IN REFS ARRAYS
      DO 10 IR=1,3
   10 NSP(IR)=NSMRB(IR)
C
C***  SET UP + COMPUTE SECTIONS (METHANE - PROPANE)
      NL2=0
      DO 20 JR=1,3
      IR=JR
      IF(LRF(IR).EQ.0) GO TO 20
      LLV(1,IR)=NL1=NL2+1
      LLV(2,IR)=NL2=NL1+LRF(IR)-1
      ICALC=0
      NC=NCMP(IR)
      TCOND=TRFC(IR)+APRR
      GO TO 100
   15 IF(IR.EQ.3) GO TO 20
C     STORE CONDENSING LOAD DATA (METHANE,ETHYLENE)
      FLOC(IR)=FLO
      QRC(IR)=QCOND
C     ADD COND LOAD TO LOWEST LEVEL LOAD FOR NEXT REFR
      QLEV(NL2+1)=QLEV(NL2+1)+QCOND
C
C*    TRANSFER RESIDUALS OF PURCH STREAMS TO NEXT SECTION
      NNSP=NSP(IR)
      IF(NNSP.EQ.0) GO TO 20
```

```
      ICALC=0
      IR=JR+1
      DO 16 JS=1,NNSP
C     LOAD SOLD STREAM RESIDUALS INTO SIN(1) , THEN INTO SMRB
      NN=NRSS(JS)
      CALL MVFSM(1,NN,JR+2,JS)
      ASSIGN 16 TO LOADS
      GO TO 400
   16 NSP(IR)=NSMRB(IR)
   20 CONTINUE
C
C***  COMPUTE TOTAL SYSTEM COSTS - INCL CPS,CPE COSTS
   25 CTOT=0.
      DO 28 L=1,NLL
   28 CTOT=CTOT+COST(L)
      DO 30 IR=1,3
   30 CTOT=CTOT+CPS(IR)+CPE(IR)
C
C     COMPUTE COSTS/BTU
      DO 32 L=1,NLL
      IF(QLEV(L).EQ.0.) GO TO 32
      IR=1
      IF(L.GE.LLV(1,2)) IR=2
      IF(L.GE.LLV(1,3)) IR=3
      COST(L)=(COST(L)/QLEV(L)+(CPS(IR)+CPE(IR))/SUMQ(IR))/HRS
   32 CONTINUE
C
C     ADD CONDENSATION COSTS
      DO 36 IR=1,2
      IF(LRF(IR).EQ.0) GO TO 36
      LEV=LLV(1,IR+1)
      QCOST=QRC(IR)*COST(LEV)
      L1=LLV(1,IR)
      L2=LLV(2,IR)
      DO 35 L=L1,L2
      IF(QLEV(L).EQ.0.) GO TO 35
      COST(L)=COST(L)+(FLEV(L)/FLOC(IR))*(QCOST/QLEV(L))
   35 CONTINUE
   36 CONTINUE
C
      WRITE(6,950)CTOT,CPS,CPE
      DO 980 IR=1,3
      IF(LRF(IR).EQ.0) GO TO 980
      KEQ=NEMR(IR)
      KSM=NSMRB(IR)
      WRITE(6,963)
      DO 970 K=1,KSM
  970 WRITE(6,972)K,(SMRB(KK,K,IR),KK=1,7)
      WRITE(6,955)
      DO 960 K=1,KEQ
  960 WRITE(6,962)(EMR(KK,K,IR),KK=1,15)
  980 CONTINUE
      WRITE(6,982)
      DO 984 L=1,NLL
  984 WRITE(6,986)TLEV(L),PLEV(L),QLEV(L),FLEV(L),COST(L)
      RETURN
C
C
C**** REFR SECTION COMPUTING ROUTINE ****
C
C     SET SIN(1) AS UNIT FLOW STREAM (SAT LIQ) AT COND PRES
  100 CALL ZERO(XIN,NOCOMP)
      TMIN(1)=XIN(NC,1)=1.
C     COMPUTE CONDENSATION PRESSURE
      TIN(1)=DPIN(1)=BPIN(1)=TCOND
      VFIN(1)=0.
      PIN(1)=P(IR,TCOND)
      WRITE(6,817)TIN(1),PIN(1)
      CALL PBUB(1,PCOND,DUM)
      PIN(1)=PLEV(NL2+1)=PCOND
      CALL ENTH(1,HIN(1),DUM)
C     LOAD INTO SMRB NLIQ
      ASSIGN 103 TO LOADS
      GO TO 400
```

```
  103 NLIQ=NSM
C
C     COMPUTE PRESSURE LEVELS + SUM QS
      DO 104 L=NL1,NL2
      TIN(1)=TLEV(L)
      PIN(1)=P(IR,TLEV(L))
      WRITE(6,817)TIN(1),PIN(1)
      CALL PBUB(1,PLEV(L),DUM)
  104 SUMQ(IR)=SUMQ(IR)+QLEV(L)
C     ESTIMATE REF CIRCULATION
      FLOW=SUMQ(IR)/DHV(IR)
C
C***  SET (RE-SET) COUNTERS ETC - START TOTAL FLOW ITERATION
  105 NSM=NSMRB(IR)=NLIQ
      NEMR(IR)=0
      ICALC=1
      CLEV=CPS(IR)=CPE(IR)=0.
      CALL ZERO(TEX,2)
      NNSP=NSP(IR)
      IF(NNSP.EQ.0) GO TO 120
C
C**   COMPUTE LIQ COOLING BY PURCHASED STREAMS
C
      NSL=NLIQ
      XFLO=FLOW
      DO 118 JP=1,NNSP
C++   ORDERING OF PURCHASED STREAMS
      NP=JP
C++
C     LOAD NSL INTO SIN(1) (SET TO FLOW FOR JP=1) , NP INTO SIN(2)
      CALL MVFSM(1,NSL,IR2,NLIQ)
      IWV=1
      ASSIGN 110 TO IFLO
      IF(JP.EQ.1) GO TO 440
  110 CALL MVFSM(2,NP,IR2,NP)
C     EXCHANGE , LOAD STREAMS + EQUIP
      JIN1=NSL
      JIN2=NP
C*
      CALL HXER(1,4,TEX,Q)
C*
      ASSIGN 112 TO LOADS
      GO TO 400
  112 NRSS(NP)=NSM
C     COMPUTE VALUE OF COOLING USED
      DT=DTF(2)*APRR
      CALL MVFSM(1,NP,IR2,NP)
      CALL SVALUE(2,TOUT(2),DT,VAL)
      CPS(IR)=CPS(IR)+VAL
  118 NSL=NSM-1
      CPE(IR)=CLEV
      GO TO 122
C
  120 NSL=NSM
C     SAVE COUNTERS
  122 NSMSV=NSM
      NEMSV=NEM
      TSL=SMRB(3,NSL,IR)
C
C***  COMPUTE FLASH (+ CROSS EXCHANGE) FOR ALL LEVELS - TO DETERMINE FLO
C
      FLO=FXO=0.
      DO 150 L=NL1,NL2
      QQ=QLEV(L)
      IF(QQ.EQ.0.) GO TO 150
      NIT=0
C     LOAD NSL INTO SIN(1)
      CALL MVFSM(1,NSL,IR2,NLIQ)
C     COMPUTE FLASH + SET UP SAT VAP OUTPUT
  124 CLEV=0.
      NIN=NOUT=1
C*
      CALL ADBF(PLEV(L))
C*
```

```
C     CALCULATE ENTHALPY CHANGE AVAIL/MOLE + REQUIRED FLOW
      VFOUT(1)=1.
      CALL ENTH(-1,HOUT(1),DUM)
      HAVL=(HOUT(1)-HIN(1))/TMIN(1)
      FXN=QQ/HAVL
C     SET SOUT(1) TO FXN
      IWV=-1
      XFLO=FXN
      ASSIGN 126 TO IFLO
      GO TO 440
C
C     LOAD STREAMS + EQUIP FOR FLASH
  126 JIN2=0
      IF(IR.EQ.3) GO TO 128
C     IS CROSS EX POSSIBLE
      DTX=(TSL-TOUT(1))-(1.5*APRR)
      IF(DTX.GT.0.) GO TO 130
C*    PROPANE
  128 JIN1=NSL
      ASSIGN 142 TO LOADS
      GO TO 400
C
C*    IR=1,2 - REQUIRES ITERATION OF  FLASH/CROSS EX  LOOP
C     RESET COUNTERS BEFORE LOADING
  130 NSM=NSMRB(IR)=NSMSV
      NEMR(IR)=NEMSV
      JIN1=NSM+2
      ASSIGN 132 TO LOADS
      GO TO 400
C     TRANSFER SOUT(1) TO SIN(2) , LOAD NSL INTO SIN(1) + SET TO FXN
  132 CALL MVSOSI(2,1,0,0)
      CALL MVFSM(1,NSL,IR2,NLIQ)
      IWV=1
      ASSIGN 134 TO IFLO
      GO TO 440
C     COMPUTE CROSS EXCHANGE , LOAD STREAMS + EQUIP
C*
  134 CALL HXER(1,4,TEX,Q)
C*
      JIN1=NSL
      JIN2=NSM
      ASSIGN 140 TO LOADS
      GO TO 400
C*    TEST FOR CONVERGENCE OF LEVEL FLOW
  140 FDF=ABS(FXN-FXO)/FXN
      IF(FDF.LT.0.01) GO TO 142
      FXO=FXN
      NIT=NIT+1
      GO TO 124
C
C     UPDATE COUNTERS
  142 NSMSV=NSM
      NEMSV=NEM
C     SET LEVEL FLOW + COST , STORE OUTPUT STREAM NO
      FLEV(L)=FXN
      FLO=FLO+FXN
      COST(L)=CLEV
      NVP(L)=NSM
  150 CONTINUE
C
C*    CHECK FOR CONVERGENCE OF TOTAL FLOW
      FDF=ABS(FLO-FLOW)/FLO
      IF(FDF.LT.0.01) GO TO 152
      FLOW=FLO
      GO TO 105
C
C     SET NSM TO NVP(NL1) + LOAD INTO SIN(1)
  152 NSM=NVP(NL1)
      CALL MVFSM(1,NSM,IR2,NLIQ)
      FLO=0.
C
C***  MIX , COMPRESS + AFTERCOOL FOR ALL LEVELS
C
      DO 200 L=NL1,NL2
```

```
      WRITE(6,510)L,TLEV(L),PLEV(L),QLEV(L),FLEV(L)
      FLO=FLO+FLEV(L)
      QQ=QLEV(L)
      CLEV=0.
      IF(L.EQ.NL1.OR.QQ.EQ.0.) GO TO 160
C**   MIX , LOAD STREAMS + EQUIP
C     LOAD NVP(L) INTO SIN(2)
      NMX=NVP(L)
      CALL MVFSM(2,NMX,IR2,NLIQ)
      NIN=2
      NOUT=1
      JIN1=NSM
      JIN2=NMX
C*
      CALL MIXR(0.)
C*
      ASSIGN 160 TO LOADS
      GO TO 400
C**   COMPRESS , LOAD STREAMS + EQUIP
  160 JIN1=NSM
      JIN2=0
C*
      CALL COMP(PLEV(L+1))
C*
      ASSIGN 166 TO LOADS
      GO TO 400
  166 IF(L.EQ.NL2.AND.IR.EQ.3) GO TO 172
C     IS AFTERCOOLING POSSIBLE
      DTA=TIN(1)-(TWAT+DTW+APPP)
      IF(DTA.LT.0.) GO TO 170
C**   AFTERCOOL , LOAD STREAMS + EQUIP
      JIN1=NSM
      JIN2=202
      TEX(1)=AMAX1((TWAT+APPP),(DPIN(1)+1.))
C*
      CALL HXER(1,2,TEX,Q)
C*
      ASSIGN 170 TO LOADS
      GO TO 400
  170 IF(L.LT.NL2) GO TO 180
C
C**   CONDENSE - LOAD EQUIP ONLY
  172 TEX(1)=BPIN(1)-1.
      IF(IR.LT.3) GO TO 173
      GO TO 180
C++   SUBTRACT PSEUDO-SERVICE REQUIREMENT
      XFLO=TMIN(1)-TMSERV
C++
      IWV=1
      IS2=2
      ASSIGN 174 TO IFLO
      GO TO 440
  173 IS2=3
      TEX(2)=TLEV(NL2+1)
  174 JIN1=NSM
      JIN2=IS2+200
C*
      CALL HXER(1,IS2,TEX,Q)
C*
      JO=0
      EQUIP(5)=NLIQ
      ASSIGN 176 TO LOADS
      GO TO 420
  176 QCOND=ABS(Q)
C
C     PROPORTION + SUM COSTS
  180 DO 182 LL=NL1,L
  182 COST(LL)=COST(LL)+CLEV*(FLEV(LL)/FLO)
      WRITE(6,990)CLEV,(COST(II),II=NL1,L)
  200 CONTINUE
      GO TO 15
C
C
C***  OUTPUT STREAM LOADING ROUTINE ***
```

HR

```
C
  400 IR2=IR+2
      IF(ICALC.EQ.1) GO TO 402
      JO=IOS=1
      GO TO 404
  402 IOS=-1
      IOP=EQUIP(2)
      JO=2
      IF(JIN2.EQ.0.OR.JIN2.GT.200.OR.IOP.EQ.21) JO=1
  404 DO 410 J=1,JO
      NSM=NSMRB(IR)=NSMRB(IR)+1
      JJ=J*IOS
  410 CALL MVTSM(JJ,NSM,IR2,0)
      IF(ICALC.EQ.1) GO TO 420
C     STORE MOLE FRACTIONS IN SMRX
      DO 412 I=1,NOCOMP
  412 SMRX(I,NSM,IR)=XIN(I,1)/TMIN(1)
      GO TO 430
C
C***  EQUIP LOADING ROUTINE
C
  420 NEM=NEMR(IR)=NEMR(IR)+1
      EQUIP(1)=NEM
      EQUIP(3)=JIN1
      EQUIP(4)=JIN2
      IF(JO.EQ.0) GO TO 422
      EQUIP(5)=NSM-(JO-1)
      IF(JO.EQ.2) EQUIP(6)=NSM
  422 DO 425 K=1,15
  425 EMR(K,NEM,IR)=EQUIP(K)
C     RESTORE SOUT(1) TO SIN(1)
      CALL MVSOSI(1,1,0,0)
      CLEV=CLEV+EQUIP(15)
  430 GO TO LOADS,(16,103,112,132,140,142,160,170,176)
C
C***  ROUTINE TO SET STREAM VECTOR IWV TO XFLO ***
C
  440 IF(IWV.GT.0) GO TO 444
      IWV=-IWV
      HOUT(IWV)=HOUT(IWV)*(XFLO/TMOUT(IWV))
      TMOUT(IWV)=XOUT(NC,IWV)=XFLO
      GO TO 450
  444 HIN(IWV)=HIN(IWV)*(XFLO/TMIN(IWV))
      TMIN(IWV)=XIN(NC,IWV)=XFLO
  450 GO TO IFLO,(110,126,134,174)
C
  510 FORMAT(//* L T,P,Q,F -*,I5,5X,2F8.1,F10.0,F8.1/)
  817 FORMAT(* T,P*,2F10.1)
  950 FORMAT(///* COSTS - CTOT,CPS,CPE*,F10.0,5X,3F10.0,5X,3F10.0//)
  955 FORMAT(/* EQUIP MATRIX-*/)
  962 FORMAT(3(2F5.0,3X),F12.0,5F9.1,3F9.0)
  963 FORMAT(/* STREAM MATRIX-*/)
  972 FORMAT(I5,3F8.1,5X,F8.1,F7.3,F8.1)
  982 FORMAT(//* T,P,Q,F,C -*/)
  986 FORMAT(2F8.1,F10.0,F8.1,F12.8)
  990 FORMAT(/* CLEV,COST*,F10.0,5X,6F8.0)
      END
```

```
      SUBROUTINE STMOVR(IWV,ISM,III,NX)
C
C     STREAM MOVING UTILITY ROUTINE ...(RUNIT VERSION)
C
C     IWV   - ELEMENT NUMBER IN SIN OR SOUT WORKING ARRAY
C             + SIN
C             - SOUT
C     ISM   - VECTOR NUMBER IN SM--
C     III   - 0     MOVE TO OR FROM SMPB
C             1-2   MOVE TO OR FROM SMCHB - (1-2)
C             3-5   MOVE TO OR FROM SMRB - (1-3)
C     NX    - VECTOR NUMBER CONTAINING MOLE FRACTIONS   (III GT 0)
C
C     3 ENTRIES -
C        1 MVSOSI MOVES SOUT VECTOR ISM TO SIN VECTOR IWV (III=0)
C        2 MVFSM MOVES FROM SM-- TO SIN OR SOUT
C        3 MVTSM MOVES TO SM-- FROM SIN OR SOUT
C
C     *** COMMON DECK ***
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
C$    BLANK COMMON
      COMMON NSMRB(3),SMRB(7,20,3),SMRX(8,4,3)
C     ***
      DIMENSION SMPB(1,1),SMCHB(1,1,1),SMCHX(1,1,1)
      DIMENSION SIDUM(4,7),SODUM(4,7)
      EQUIVALENCE (BPIN,SIDUM),(BPOUT,SODUM)
C
C**
      ENTRY MVSOSI
      IENT=1
      GO TO 1
C**
      ENTRY MVFSM
      IENT=2
      GO TO 1
C**
      ENTRY MVTSM
      IENT=3
C
    1 JJJ=III+1
      GO TO (2,3,3,4,4,4)JJJ
    2 ITYPE=1
      GO TO 5
    3 ITYPE=2
      KKK=III
      GO TO 5
    4 ITYPE=3
      KKK=III-2
C
    5 GO TO (100,200,300)IENT
C
C**   MVSOSI
  100 DO 50 I=1,7
   50 SIDUM(IWV,I)=SODUM(ISM,I)
      DO 60 I=1,NOCOMP
   60 XIN(I,IWV)=XOUT(I,ISM)
      RETURN
C
C**   MVFSM
  200 DO 10 I=1,7
      GO TO (6,7,8)ITYPE
    6 AA=SMPB(I,ISM)
      GO TO 9
    7 AA=SMCHB(I,ISM,KKK)
      GO TO 9
    8 AA=SMRB(I,ISM,KKK)
    9 IF(IWV.GT.0) SIDUM(IWV,I)=AA
      IF(IWV.LT.0) SODUM(-IWV,I)=AA
   10 CONTINUE
C
      DO 20 I=1,NOCOMP
      GO TO (11,12,13)ITYPE
```

```
   11 AA=SMPB(I+7,ISM)
      GO TO 18
   12 AA=SMCHX(I,NX,KKK)*SMCHB(7,ISM,KKK)
      GO TO 18
   13 AA=SMRX(I,NX,KKK)*SMRB(7,ISM,KKK)
   18 IF(IWV.GT.0) XIN(I,IWV)=AA
      IF(IWV.LT.0) XOUT(I,-IWV)=AA
   20 CONTINUE
      RETURN
C
C**   MVTSM
  300 DO 30 I=1,7
      IF(IWV.GT.0) AA=SIDUM(IWV,I)
      IF(IWV.LT.0) AA=SODUM(-IWV,I)
      GO TO (21,22,23)ITYPE
   21 SMPB(I,ISM)=AA
      GO TO 30
   22 SMCHB(I,ISM,KKK)=AA
      GO TO 30
   23 SMRB(I,ISM,KKK)=AA
   30 CONTINUE
C
      IF(ITYPE.GT.1) RETURN
      DO 40 I=1,NOCOMP
      IF(IWV.GT.0) AA=XIN(I,IWV)
      IF(IWV.LT.0) AA=XOUT(I,-IWV)
   40 SMPB(I+7,ISM)=AA
      RETURN
      END
```

E PHYSICAL PROPERTIES

PROPL is the routine which contains the library of basic pure component physical properties constants. It punches out data decks for any selected set of components; such data is then read into the /PROP/ labelled COMMON arrays in KHZT for use in properties correlations.

KHZT is, as described in section 4.2.5, the physical properties calculation routine which supplies equilibrium, enthalpy and density data to other system routines. Apart from some nomenclature changes and streamlining and the modifications described in chapter 8, it is the CHESS[2] physical properties package and a detailed description of its function can be found in that reference.

ZERO and ZEROI are not actually property routines but are included here as they are also system service routines. They provide automatic (floating point and integer) array zeroing.

FLASH serves both as part of the properties system and as an equipment subroutine. It provides a rigorous two-phase calculation capability for other routines, either in isothermal or adiabatic mode (entries ISOF and ADBF). The isothermal mode is essentially a direct calculation which computes the enthalpy of the outlet stream (a two-phase stream or separate liquid and vapor streams, as specified by the user) for a specified inlet stream temperature. In the adiabatic mode an iterative isothermal calculation is generally required to establish the outlet stream(s) temperature corresponding to the specified inlet stream enthalpy. As with KHZT, FLASH is derived from the corresponding CHESS[2] system routine (ADBF) and modifications are again described in chapter 8.

As an equipment subroutine FLASH represents either adiabatic (valve) expansion (entry ADBF) or adiabatic mixing (entry MIXR).

```
      PROGRAM PROPL(INPUT,OUTPUT,PUNCH,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7=PU
     1NCH)
C      PHYSICAL POPERTIES DATA LIBRARY ROUTINE
C      PUNCHES OUT DATA FOR SELECTED COMPONENTS
C
C
C     PURE COMPONENT ID NUMBERS...
C
C      1. HYDROGEN         18. N-TETRADECANE       35. 1-HEXENE
C      2. METHANE          19. N-PENTADECANE       36. CYCLOPENTANE
C      3. ETHANE           20. N-HEXADECANE        37. METHYLCYCLOPENTANE
C      4. PROPANE          21. N-HEPTADECANE       38. CYCLOHEXANE
C      5. I-BUTANE         22. ETHYLENE            39. METHYLCYCLOHEXANE
C      6. N-BUTANE         23. PROPYLENE           40. BENZENE
C      7. I-PENTANE        24. 1-BUTENE            41. TOLUENE
C      8. N-PENTANE        25. CIS-2-BUTENE        42. O-XYLENE
C      9. NEO-PENTANE      26. TRANS-2-BUTENE      43. M-XYLENE
C     10. N-HEXANE         27. I-BUTENE            44. P-XYLENE
C     11. N-HEPTANE        28. 1,3-BUTADIENE       45. ETHYLBENZENE
C     12. N-OCTANE         29. 1-PENTENE           46. NITROGEN
C     13. N-NONANE         30. CIS-2-PENTENE       47. OXYGEN
C     14. N-DECANE         31. TRANS-2-PENTENE     48. CARBON MONOXIDE
C     15. N-UNDECANE       32. 2-METHYL-1-BUTENE 49. CARBON DIOXIDE
C     16. N-DODECANE       33. 3-METHYL-1-BUTENE 50. HYDROGEN SULFIDE
C     17. N-TRIDECANE      34. 2-METHYL-2-BUTENE 51. SULFUR DIOXIDE
C
C     52. 2-METHYL-C5      56. 1-HEPTENE           60. C2-CYCLO-C6
C     53. 3-METHYL-C5      57. PROPADIENE          61. ISOPRENE
C     54. 2,2-DI-C1-C4     58. 1,2-BUTADIENE       62. WATER
C     55. 2,3-DI-C1-C4     59. C2-CYCLO-C5
C
      COMMON/PROP/APC(10),ATC(10),AVC(10),AMW(10),AOMEG(10),ADEL(10),
     1AVW(10),APH(10),BET(10),GAM(10),DTA(10),
     2BASEA(10),BASEB(10),ZCD(10),ALD(10)
C
C      STANDARD COMPONENT NAMES
C
      INTEGER SCNAME(248)
      DATA (SCNAME(I),I=1,156)/
     1            4H HYD,4HROGE,4HN   ,4H    ,4H MET,4HHANE,4H    ,4H
     1 ,4H ETH,4HANE ,4H    ,4H    ,4H PRO,4HPANE,4H    ,4H    ,4H I-B,4
     2HUTAN,4HE   ,4H    ,4H N-B,4HUTAN,4HE   ,4H    ,4H I-P,4HENTA,4HNE
     3  ,4H    ,4H N-P,4HENTA,4HNE  ,4H    ,4H NEO,4H-PEN,4HTANE,4H    ,
     44H N-H,4HEXAN,4HE   ,4H    ,4H N-H,4HEPTA,4HNE  ,4H    ,4H N-O,4HC
     5TAN,4HE   ,4H    ,4H N-N,4HONAN,4HE   ,4H    ,4H N-D,4HECAN,4HE
     6,4H    ,4H N-U,4HNDEC,4HANE ,4H    ,4H N-D,4HODEC,4HANE ,4H    ,4H
     7 N-T,4HRIDE,4HCANE,4H    ,4H N-T,4HETRA,4HDECA,4HNE  ,4H N-P,4HENT
     8A,4HDECA,4HNE  ,4H N-H,4HEXAD,4HECAN,4HE   ,4H N-H,4HEPTA,4HDECA,4
     8HNE  ,4H ETH,4HYLEN,4HE   ,4H    ,
     9  4H PRO,4HPYLE,4HNE  ,4H    ,4H 1-B,4HUTEN,4HE   ,4H    ,4H CIS,
     A4H-2-B,4HUTEN,4HE   ,4H TRA,4HNS-2,4H-BUT,4HENE ,4H I-B,4HUTEN,4HE
     B  ,4H    ,4H 1,3,4H-BUT,4HADIE,4HNE  ,4H 1-P,4HENTE,4HE   ,4H
     C,4H CIS,4H-2-P,4HENTE,4HNE  ,4H TR-,4H2-PE,4HNTEN,4HE   ,4H 2-C,4H
     D1-1-,4HBUTE,4HNE  ,4H 3-C,4H1-1-,4HBUTE,4HNE  ,4H 2-C,4H1-2-,4HBUT
     EE,4HNE  ,4H 1-H,4HEXEN,4HE   ,4H    ,4H CYC,4HLOPE,4HNTEN,4HE   ,4
     FH C1-,4HCYCL,4HO-C5,4H    ,4H CYC,4HLOHE,4HXANE,4H    ,4H C1-,4HCY
     GCL,4HO-C6,4H    /
      DATA (SCNAME(I),I=157,248)/
     G            4H BEN,4HZENE,4H    ,4H    ,4H TOL,4HUENE,4H    ,4H    ,
     H4H O-X,4HYLEN,4HE   ,4H    ,4H M-X,4HYLEN,4HE   ,4H    ,4H P-X,4HY
     ILEN,4HE   ,4H    ,4H ETH,4HYLBE,4HNZEN,4HE   ,4H NIT,4HROGE,4HN
     J,4H    ,4H OXY,4HGEN ,4H    ,4H    ,4H CO ,4H    ,4H    ,4H    ,4H
     K CO2,4H    ,4H    ,4H    ,4H H2S ,4H    ,4H    ,4H    ,4H SO2, 4H
     L    ,4H    ,4H    ,4H 2-M,4HETHY,4HL-C5,4H    ,4H 3-M,4HETHY,4HL-C5
     M,4H    ,4H 2,2,4H-DI-,4HC1-C,4H4    ,4H 2,3,4H-DI-,4HC1-C,4H4    ,4H
     N 1-H,4HEPTE,4HNE  ,4H    ,4H PRO,4HPADI,4HENE ,4H    ,4H 1,2,4H-BU
     OT,4HADIE,4HNE  ,4H C2-,4HCYCL,4HO-C5,4H    ,4H C2-,4HCYCL,4HO-C6,4
     PH    ,4H ISO,4HPREN,4HE   ,4H    ,4H WAT,4HER  ,4H    ,4H    /
C
C      CHAO-SEADER MODIFIED ACENTRIC FACTORS - DIMENSIONLESS
      REAL OMEGA(62)
      DATA OMEGA/     2*0.,.1064,.1538,.1825,.1953,.2014,.2387,.195,.2927
     1,.3403,.3992,.4439,.4869,.5210,.5610,.6002,.6399,.6743,.7078,.7327
     2,.0949,.1451,.2085,.2575,.2230,.1975,.2028,.2198,.2060,.2090,.2000
     3,.1490,.2120,.2463,.2051,.2346,.2032,.2421,.2130,.2591,.2904,.3045
     4,.2969,.2936,.0206,.0299,-.0067,.1768,.0868,.2402,.2771,.2746,.231
```

```
54    ,.2466,.3471,.1193,.0987,.2709,.3046,.213,.348/
C
C      CHAO-SEADER MODIFIED HILDEBRAND SOLUBILITY PARAMETER
C           (CAL./ML.)** 1/2
       REAL DEL(62)
       DATA DEL/      3.25,5.45,5.88,6.00,2*6.73,3*7.021,7.266,7.43,7.551,7
      1.649,7.721,7.79,7.84,7.89,7.92,7.96,7.99,8.03,5.8,6.2,4*6.76,6.94,
      26*7.055,7.4,8.107,7.849,8.196,7.826,9.158,8.915,8.987,8.818,8.769,
      38.787,2.58,4.,3.13,6.,5.634,6.,7.018,7.132,6.712,6.967,7.168,6.854
      4,7.95,7.739,7.743,7.277,7.39/
C
C      VOLUME AT 25 DEG.C. , ML./ G-MOLE
       REAL V25(62)
       DATA V25/      31.,52.,68.,84.,105.5,101.4,117.4,116.1,123.3,136.6,
      1147.5,163.5,179.6,196.,212.2,228.6,244.9,261.3,277.8,294.1,310.4,
      261.,79.,95.3,91.2,93.8,95.4,88.,110.4,107.8,109.,108.7,112.8,106.7
      3,125.8,94.7,113.1,108.7,128.3,89.4,106.8,121.2,123.5,124.,123.1,
      4 36.0,28.4,35.2,53.6,43.6,45.2,132.9,130.6,122.7,131.2,141.7,61.6,
      5 83.7,128.8,143.1,100.37,18.076/
C
C      CHAO-SEADER CHARACTERISTIC MOLAR VOLUMES - ML./ G-MOLE
       REAL VW(62)
       DATA VW/      .955,5.,7.88,10.35,13.37,13.,15.36,15.27,15.89,17.64,2
      10.05,22.49,24.94,27.42,29.9,32.39,34.88,37.39,39.89,42.41,44.92,6.
      288,9.69,12.17,11.71,12.,12.17,11.27,14.55,14.26,14.41,14.31,14.77,
      314.14,16.9,12.72,15.33,14.87,17.67,12.26,14.83,17.03,17.28,17.34,1
      47.23,2.534,2.871,2.584,6.365,5.081,6.516,17.727,17.473,16.297,
      517.519,19.223,7.721,10.936,17.713,19.916,13.297,2.552/
C
C      CRITICAL TEMPERATURES, DEG. K.
       REAL TC(62)
       DATA TC/      33.27,190.7,305.43,369.97,408.14,425.17,461.,469.78,43
      13.76,507.9,540.16,569.4,595.,619.,640.,659.,677.,695.,710.,725.,73
      25.,283.06,365.1,419.6,428.,428.,417.89,425.,474.,481.16,479.16,472
      3.16,461.16,477.16,503.99,511.76,532.77,553.46,572.16,562.61,594.,6
      432.2,619.2,618.2,619.7,126.2,154.8,81.7,194.7,211.4,263.2,498.06,
      5504.33,489.39,500.28,535.5,392.78,458.06,569.44,602.61,484.28,
      6647.33/
C
C      CRITICAL PRESSURES, ATM.
       REAL PC(62)
       DATA PC/      12.79,45.8,48.2,42.01,36.,37.47,32.9,33.31,31.57,29.92
      1,27.01,24.64,22.5,20.8,19.2,17.9,17.,16.,15.,14.,13.,50.5,45.4,39.
      27,41.,41.,39.45,42.7,39.9,35.3,35.1,35.,34.5,35.9,32.1,44.55,37.36
      3,38.2,34.32,48.6,40.,36.,35.,34.,37.,33.5,50.1,34.5,72.9,88.9,77.7
      4,29.94,30.83,30.65,30.99,28.05,45.92,40.12,33.53,30.88,38.,218.37/
C
C      CRITICAL VOLUMES,  CC./GMOLE
       REAL VC(62)
       DATA VC/      65.0,99.5,148.,200.,263.,255.,308.,311.,303.,368.,426.
      1,486.,543.,602.,660.,718.,780.,830.,890.,950.,1000.,124.,181.,240.
      2,236.,240.,235.,221.,295.,295.,295.,301.,291.,286.,350.,260.,319.,
      3308.,344.,260.,316.,369.,376.,378.,374.,90.1,74.4,93.1,94.,95.,122
      4.,367.,367.,359.,358.,405.,146.,221.,375.,419.,266.,56./
C
C      MOLECULAR WEIGHTS
       REAL MW(62)
       DATA MW/      2.016,16.042,30.068,44.094,2*58.12,3*72.146,86.172,
      1100.198,114.224,128.25,142.276,156.302,170.328,184.354,198.38,
      2212.406,226.432,240.458,28.052,42.078,4*56.104,54.088,6*70.13,
      384.156,70.13,2*84.156,98.182,78.108,92.134,4*106.16,28.016,32.,28.
      401,44.01,34.08,64.06,4*86.2,98.2,40.1,54.1,98.2,112.2,68.1,18.02/
C
C      DENSITIES AT 15 DEG. C.,  G./ML.
       REAL DENS(62)
       DATA DENS/      .07,.2,.376,.5076,.5633,.5847,.6246,.63089,.5967,.66
      1384,.68801,.70654,.72146,.7339,.7440,.7525,.7600,.7663,.7720,.7734
      2,.7780,.3490,.5226,.6014,.6271,.61,.6005,.6274,.64565,.6607,.6534,
      3.6558,.6326,.6776,.67779,.75018,.7534,.78314,.77371,.88417,.87146,
      4.88440,.86836,.86532,.87141,.808,1.140,.804,1.101,.790,1.434,.6579
      5,.669,.654,.6664,.7015,.657,.658,.771,.7922,.6861,1.0/
C
C      **** COEFFICIENTS OF ZERO PRESSURE HEAT CONTENT. *****
       REAL APHA(62)
```

```
      DATA APHA/     6.952,3.381,2.247,2.410,3.332,4.453,4.816,5.910,4.37
     17,7.477,9.055,10.626,12.198,13.770,15.342,16.914,18.486,20.064,21.
     263,23.202,24.774,.944,.753,-.24,-1.778,2.34,1.65,-1.29,1.788,-3.35
     31,1.49,.495,3.270,.130,2.063,-12.957,-12.114,-15.935,-15.07,-8.65,
     4-8.213,-3.789,-6.533,-5.334,-8.398,6.903,6.085,6.726,5.316,7.07,6.
     5157,1.361,2.621,.593,1.298,2.344,3.0159,2.8487,-12.282,-15.559,
     6.4687,7.70/
      REAL BETTA(62)
      DATA BETTA/      -.04576E-2,18.044E-3,38.201E-3,57.195E-3,75.214E-3,
     172.27E-3,91.585E-3,88.449E-3,94.61E-3,104.422E-3,120.352E-3,136.29
     28E-3,152.248E-3,168.198E-3,184.148E-3,200.098E-3,216.048E-3,231.99
     37E-3,247.948E-3,263.898E-3,279.848E-3,3.735E-2,5.691E-2,8.65E-2,8.
     4078E-2,7.22E-2,7.702E-2,8.35E-2,101.454E-3,109.623E-3,99.696E-3,10
     53.985E-3,99.735E-3,99.118E-3,123.004E-3,13.087E-2,15.380E-2,16.454
     6E-2,18.972E-2,11.578E-2,13.357E-2,14.291E-2,14.905E-2,14.220E-2,15
     7.935E-2,-.03753E-2,.3631E-2,.04001E-2,1.4285E-2,.3128E-2,1.384E-2,
     812.5712E-2,12.3504E-2,13.3E-2,12.6929E-2,14.4802E-2,4.503E-2,
     9 6.4329E-2,17.682E-2,21.3801E-2,9.487E-2,4.594E-4/
      REAL GAMA(62)
      DATA GAMA/       .09563E-5,-43.E-7,-110.49E-7,-175.33E-7,-237.34E-7,-
     1222.14E-7,-289.62E-7,-273.88E-7,-305.87E-7,-324.71E-7,-375.28E-7,-
     2425.93E-7,-476.62E-7,-527.31E-7,-578.E-7,-628.69E-7,-679.38E-7,-73
     30.02E-7,-780.76E-7,-831.45E-7,-882.14E-7,-1.993E-5,-2.91E-5,-5.11E
     4-5,-4.074E-5,-3.403E-5,-3.981E-5,-5.582E-5,-554.27E-7,-603.45E-7,-
     5582.63E-7,-574.04E-7,-551.51E-7,-504.37E-7,-674.01E-7,-7.447E-5,-8
     6.915E-5,-9.203E-5,-10.989E-5,-7.54E-5,-8.23E-5,-8.354E-5,-8.831E-5
     7,-7.984E-5,-10.003E-5,.1930E-5,-.1709E-5,.1283E-5,-.8362E-5,.1364E
     8-5,-.9103E-5,-4.8147E-5,-4.7104E-5,-5.2878E-5,-4.9133E-5,-7.9864E-
     95,-2.556E-5,-3.418E-5,-10.2304E-5,-12.3808E-5,-5.553E-5,2.521E-6/
      REAL DELTA(62)
      DATA DELTA/      -.2079E-9,20*0.,4.22E-9,5.88E-9,12.07E-9,7.89E-9,6.
     107E-9,8.02E-9,14.24E-9,11.855E-9,12.911E-9,10.948E-9,12.414E-9,12.
     2041E-9,9.9E-9,14.404E-9,16.41E-9,20.03E-9,19.27E-9,24.09E-9,18.54E
     3-9,19.20E-9,18.8E-9,20.05E-9,17.03E-9,23.95E-9,-.6861E-9,.3133E-9,
     4-.5307E-9,1.784E-9,-.7867E-9,2.057E-9,4*0.,17.18E-9,5.678E-9,7.043
     5E-9,22.835E-9,26.997E-9,1.2629E-8,-8.587E-10/
C
C
C     READ COMPONENT NUMBER
  800 READ(5,712)I
      IF(EOF,5)720,721
  720 CALL EXIT
  721 I1=4*(I-1)+1
      I2=I1+3
      VC(I)=VC(I)*.45359/28.32
C     PUNCH DATA
      WRITE(7,712)I,(SCNAME(JJ),JJ=I1,I2)
      PP=14.696*PC(I)
      TT=1.8*TC(I)
      BASEB(I)=0.0867*TT/PP
      ZCD(I)=PP*VC(I)/(10.73*TT)
      BASEA(I)=SQRT(0.4278*TT**2.5/PP)
      WRITE(7,714)PP,TT,VC(I),MW(I),OMEGA(I),DEL(I),VW(I),APHA(I),BETTA(
     1I),GAMA(I),DELTA(I),BASEA(I),BASEB(I),ZCD(I),DENS(I)
      GO TO 800
C
  712 FORMAT(I5,4A4)
  714 FORMAT(5E14.5)
      END
```

```
      SUBROUTINE KHZT(ARG,ANS,LIST)
C
C    THIS IS A COMPREHENSIVE THERMO. DATA SUBROUTINE WITH 8 ENTRY POINTS
C                   1. ENTRY DENS  (ARG,ANS,AAMW)
C                   2. ENTRY ENTH  (ARG,ANS,DUM)
C                   3. ENTRY KVAL  (ARG,ANS,LIST)
C                   4. ENTRY TSUBH (ARG,ANS,DUM)
C                   5. ENTRY BUBTP (ARG,ANS,LIST)
C                   6. ENTRY DEWTP (ARG,ANS,LIST)
C                   7. ENTRY PBUB  (ARG,ANS,LIST)
C                   8. ENTRY PDEW  (ARG,ANS,LIST)
C
C
C*****   COMMON DECK
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
      COMMON/PROP/COMPNT(8),APC(8),ATC(8),AVC(8),AMW(8),AOMEG(8),
     1 ADEL(8),AVW(8),APH(8),BET(8),GAM(8),DTA(8),BASEA(8),BASEB(8),
     2 ZCD(8),ALD(8)
C*****
C
      INTEGER ARG,COUNTT,COUNT,COUN,COMPNT,VPFRAC
      REAL X(8),LIST(8),LNPHI(8),LNACT(8),LNNU(8),KV(8),NEWX(8),AV25(8)
      REAL LHC,MPOLY,MADDY
      LOGICAL FFLAG,FLAG,AKFLAG
      EQUIVALENCE (TRE,TEMTUR)
      DATA TTLOW/240./
C
C     INTERNAL FUNCTIONS        ***** DELHV + DELHL *****
C     AS STATEMENT FUNCTION    DELHVL
C
      DELHVL(H,Z)=(1.5*ASQDB*ALOG(1.+H)+1.-Z)*TEMTUR*1.986
C
      DPOLY(A,B,F,H,Z)=A+(B+(F+H*Z)*Z)*Z
      DADDY(A,B,F,H,Z)=A+B*SRED1+F*SRED2+H*ARED+Z*SRED4
      MPOLY(A1,A2,A3,A4,Z)=A1+(A2+(A3+A4*Z)*Z)*Z
      MADDY(A1,A2,A3,A4,A5)=A1+A2*SRED1+A3*SRED2+A4*ARED+A5*SRED4
C
C     FUNCTION FOR TEMTUR STARTING VALUE FOR BUBBLE + DEW POINT ITERATION
C     REGRESSED OVER RANGE  AAMW=11,55 , PRSSUR=10,565 (HYDROCARBONS CO-C
      TEMST(AAMW,PRSSUR)=-42.0+AAMW*(18.6-0.143*AAMW)+0.282*PRSSUR
     1 -1007./PRSSUR
C
C     CHAO-SEADER COEFFICIENTS FOR LIQUID FUGACITY
C     AS MODIFIED BY GRAYSON AND STREED.
C     CHAO-SEADER COEFFS RETAINED FOR H2,CH4
      REAL COEFFT(3,10)
      DATA COEFFT/1.96718,2.43840,2.05135,1.02972,-2.24550,-2.10899,
     1 -0.054009,-0.34084,0.,0.0005288,0.00212,-0.19396,0.,-0.00223,
     2 0.02282,0.008585,0.10486,0.08852,0.,-0.03691,3*0.,-0.00872,2*0.,
     3 -0.00353,2*0.,0.00203/
C
C     CONSTANTS FOR YEN AND WOODS CORRELATION
      REAL FRI(5,3)
      DATA FRI/      -.0817,.3274,-.5014,.3870,-.1342,.0933,-.3445,.4042
     1,-.2083,.05473,.089,-.4344,.7915,-.7654,.3367/
      REAL FRJ(5,3)
      DATA FRJ/      -.0230,-.0124,.1625,-.2135,.08643,.022,-.003363,
     1 -.0796,.08546,-.0217,.0674,-.06109,.06261,-.2378,.1665/
      REAL FRK(4,3)
      DATA FRK/       .05626,-.3518,.6194,-.3809,.01937,-.03055,.06310,0.0
     1,-.01393,-.003459,-.1611,0.0/
      REAL FRL(5,3)
      DATA FRL/      -21.0,55.174,-33.637,-28.109,26.277,-16.0,30.699,
     1 19.645,-81.305,47.031,-6.55,7.8027,15.344,-37.04,20.169/
C     ... PRESET VALUES ...
      DATA LNPHI,LNACT,LNNU/24*0./
C
C                              ***** ZDENS *****
C
      ENTRY DENS
      ASSIGN 20 TO LOC
```

```
      GO TO 1000
   20 ASSIGN 22 TO LOC
      GO TO 3000
   22 IF (VPFRAC .NE. 1) GO TO 24
      ASSIGN 23 TO LOC
      GO TO 5000
   23 ANS=PRSSUR/(10.73*TEMTUR*ZFAC)
      GO TO 28
   24 IF (VPFRAC .NE. 0) GO TO 26
      ASSIGN 25 TO LOC
      GO TO 6000
   25 ANS=RO/AAMW
   28 LIST(1)=AAMW
      RETURN
   26 WRITE(6,27)
   27 FORMAT(80H0*** DENSITY CANNOT BE CALCULATED BECAUSE VAPOR FRACTION
     1 IS IMPROPERLY SPECIFIED)
      ANS=0.5
      RETURN
C
C                              ***** ENTH *****
C
      ENTRY ENTH
      ASSIGN 30 TO LOC
      GO TO 1000
   31 ANS=0.
      RETURN
   30 IF(TEMTUR.LT.1.) GO TO 31
      LOS=1
      GO TO 14000
   33 ANS=GETH
      RETURN
C
C                              ***** KVAL *****
C
      ENTRY KVAL
      ASSIGN 40 TO LOC
      GO TO 1000
   40 IF(VPFRAC .NE. 0) GO TO 43
      LOS=1
      GO TO 7000
   43 IF(VPFRAC .NE. 1) GO TO 45
      LOS=1
      GO TO 8000
   45 WRITE(6,46)
   46 FORMAT(71H0*** K-VALUES CANNOT BE CALCULATED, VAPOR FRACTION IMPRO
     1PERLY SPECIFIED)
   47 DO 48 I=NFC,NLC
   48 LIST(I)=KV(I)
      RETURN
C
C                              ***** TSUBH *****
C
      ENTRY TSUBH
      ICAL=1
      ASSIGN 50 TO LOC
      GO TO 1000
   50 IF(TEMTUR.EQ.0.) TEMTUR=500.
      FRDV=0.10
      DO 56 COUNT=1,10
      LOS=2
      GO TO 14000
   52 HTRY=GETH
      SUMM=HCONT-HTRY
      IF(ABS(SUMM/HCONT) .GT. 1.E-3 ) GO TO 55
      ANS=TEMTUR
      RETURN
   55 ASSIGN 56 TO LOC
      GO TO 2000
   56 CONTINUE
      WRITE( 6,57)
   57 FORMAT(54H0*** TEMPERATURE AT INDICATED ENTHALPY CANNOT BE FOUND)
      ANS=TEMTUR
      RETURN
```

```
C
C                                        ***** BUBTP *****
C
      ENTRY BUBTP
      ICAL=1
  650 ASSIGN 60 TO LOC
      GO TO 1000
   61 ANS=0.
      WRITE(6,611)
  611 FORMAT(63H *** ABOVE CRITICAL PRESSURE - DEW/BUBBLE POINT CANNOT B
     1E FOUND)
      RETURN
   60 IF(ICAL.EQ.2) GO TO 651
C     ****
C     CALC APPROX STARTING VALUE FOR TEMTUR
      TEMTUR=TEMST(AAMW,PRSSUR)-15.
      FRDV=0.07
C     ****
  651 IF(COUNT.EQ.1) GO TO 70
      LOS=1
      GO TO 4000
   63 IF(PRSSUR.GT.(1.05*PCRIT)) GO TO 61
      ENPHI=0.
      CALL ZERO(LNPHI,NOCOMP)
      CALL ZERO(NEWX,NOCOMP)
      DO 164 COUNT=1,15
      ASSIGN 64 TO LOC
      GO TO 12000
   64 ASSIGN 65 TO LOC
      GO TO 11000
   65 DO 66   I=NFC,NLC
      TEMP=NEWX(I)
      NEWX(I)=X(I)
   66 X(I)= TEMP
      COUN=0
   67 FFLAG=.FALSE.
      IF(COUNT .EQ. 1) GO TO 68
      LOS=2
      GO TO 8000
   68 ASSIGN 69 TO LOC
      GO TO 9000
   69 DO 160 I=NFC,NLC
      TEMP=KV(I)*NEWX(I)
      IF(ABS((TEMP-X(I))/TEMP).GT. 1.E-5) FFLAG=.TRUE.
  160 X(I)=TEMP
      COUN=COUN+1
      IF( COUN.GT.20) GO TO 165
      IF(FFLAG) GO TO 67
      SUMM=0.
      DO 161 I=NFC,NLC
      TEMP=X(I)
      X(I)=NEWX(I)
      NEWX(I)=TEMP
      SUMM=SUMM+(1.-KV(I))*X(I)
  161 CONTINUE
      IF(ABS(SUMM)/TMOLE .GT. 5.E-3 ) GO TO 163
 1611 DO 162 I=NFC,NLC
  162 LIST(I)=KV(I)
      IF(ICAL.EQ.2) GO TO IPR,(310,410)
      ANS=TEMTUR
      RETURN
  163 ASSIGN 164 TO LOC
      GO TO 2000
  164 CONTINUE
  165 WRITE(  6,166)
  166 FORMAT(38H0*** BUBBLE POINT CANNOT BE DETERMINED)
      GO TO 1611
C
C                                        ***** DEWTP *****
C
      ENTRY DEWTP
      ICAL=1
  750 ASSIGN 70 TO LOC
      GO TO 1000
```

```
   70 LOS=2
      IF(ICAL.EQ.2) GO TO 751
C     ****
C     CALC APPROX STARTING VALUE FOR TEMTUR
      TEMTUR=TEMST(AAMW,PRSSUR)
      FRDV=0.07
C     ****
  751 GO TO 4000
   73 IF(PRSSUR.GT.(1.05*PCRIT)) GO TO 61
      TT=TEMTUR
      ENACT=0.
      CALL ZERO(LNACT,NOCOMP)
      CALL ZERO(NEWX,NOCOMP)
      DO 174 COUNT=1,15
      ASSIGN 74 TO LOC
      GO TO 3000
   74 ASSIGN 75 TO LOC
      GO TO 5000
   75 ASSIGN 76 TO LOC
      GO TO 10000
   76 DO 77 I=NFC,NLC
      TEMP= NEWX(I)
      NEWX(I)= X(I)
   77 X(I)=TEMP
      COUN=0
   78 FFLAG=.FALSE.
      LOS=2
      GO TO 7000
   79 DO 170 I=NFC,NLC
      TEMP=NEWX(I)/KV(I)
      IF(ABS((TEMP-X(I))/TEMP).GT. 1.E-5 ) FFLAG=.TRUE.
  170 X(I)=TEMP
      COUN=COUN+1
      IF( COUN.GT.20) GO TO 175
      IF( FFLAG) GO TO 78
      SUMM=0.
      DO 171 I=NFC,NLC
      TEMP=X(I)
      X(I)=NEWX(I)
      NEWX(I)=TEMP
      SUMM=SUMM+(1./KV(I)-1.)*X(I)
  171 CONTINUE
      IF(ABS(SUMM)/TMOLE .GT. 5.E-3 ) GO TO 173
      GO TO 1611
  173 ASSIGN 174 TO LOC
      GO TO 2000
  174 CONTINUE
  175 WRITE(  6,176)
  176 FORMAT(35H0*** DEW POINT CANNOT BE DETERMINED)
      GO TO 1611
C
C                                   ***** PBUB  *****
C
      ENTRY PBUB
      ICAL=2
      FRDV=-0.25
      ASSIGN 310 TO IPR
      GO TO 650
  310 ANS=PRSSUR
      RETURN
C
C                                   ***** PDEW  *****
C
      ENTRY PDEW
      ICAL=2
      FRDV=-0.25
      ASSIGN 410 TO IPR
      GO TO 750
  410 ANS=PRSSUR
      RETURN
C
C     INTERNAL FUNCTION             ***** COMVEC *****
C
 1000 IF(ARG.GT.0) GO TO 1007
```

```
      J=-ARG
      DO 1006 I=1,NOCOMP
 1006 X(I)=XOUT(I,J)
      TEMTUR=TOUT(J)
      PRSSUR=POUT(J)
      HCONT=HOUT(J)
      VPFRAC=VFOUT(J)+0.001
      TMOLE=TMOUT(J)
      GO TO 1009
 1007 J=ARG
      DO 1008 I=1,NOCOMP
 1008 X(I)=XIN(I,J)
      TEMTUR=TIN(J)
      PRSSUR=PIN(J)
      HCONT=HIN(J)
      VPFRAC=VFIN(J)+0.001
      TMOLE=TMIN(J)
 1009 COUNT=0
      NLC=0
      AAMW=0.
      DO 1005 I=1,NOCOMP
      XX=X(I)/TMOLE
      IF(XX.LT.0.002) GO TO 1005
      AAMW=AAMW+AMW(I)*XX
      COUNT=COUNT+1
      IF(NLC.GT.0) GO TO 1004
      NFC=NLC=I
      GO TO 1005
 1004 NLC=I
 1005 CONTINUE
      NCMP=COUNT
      GO TO LOC,(20,30,40,50,60,70)
C
C     INTERNAL FUNCTION        ***** ITER *****
C
C     ICAL=1  ITERATING ON TEMTUR
C     ICAL=2  ITERATING ON PRSSUR
C
 2000 IF(ICAL.EQ.1) VAR=TEMTUR
      IF(ICAL.EQ.2) VAR=PRSSUR
      OLVAR=VAR
      IF(COUNT.GT.1) GO TO 2001
C
C     SUMM     + TEMTUR TOO LOW , PRSSUR TOO HIGH
C              - TEMTUR TOO HIGH , PRSSUR TOO LOW
C
      IF(SUMM)2005,2005,2003
 2001 IF(II)2002,2008,2004
 2002 IF(SUMM.LT.0.) GO TO 2005
      II=0
 2003 VLOW=VAR
      SUML=SUMM
      IF(COUNT.EQ.1) GO TO 2007
      IF(II.EQ.1) GO TO 2015
      GO TO 2012
 2007 II=1
      GO TO 2014
 2004 IF(SUMM.GT.0.) GO TO 2003
      II=0
 2005 VHIGH=VAR
      SUMH=SUMM
      IF(COUNT.EQ.1) GO TO 2006
      IF(II.EQ.-1) GO TO 2015
      GO TO 2012
 2006 II=-1
      GO TO 2014
C
 2008 IF(SUMM.GT.0.) GO TO 2010
      VHIGH=VAR
      SUMH=SUMM
      GO TO 2012
 2010 VLOW=VAR
      SUML=SUMM
C
```

```
 2012 VAR=(VLOW*SUMH-VHIGH*SUML)/(SUMH-SUML)
      DIFV=VHIGH-VLOW
      IF(NCMP.EQ.1) GO TO 2113
      SGN=SIGN(1.,FRDV)
      DFV=DIFV*SGN
      IF(ICAL.EQ.1) VTOL=2.
      IF(ICAL.EQ.2) VTOL=0.02*VAR
      IF(DFV.GT.VTOL) GO TO 2113
C**
C     SPECIAL RESTART PROCEDURE FOR BUBTP + PBUB CONVERGENCE
C     RE-ESTABLISHES LOW TEMP OR HIGH PRES (LOW K VALUE) LIMIT
      II=-1
      DV=-0.5*FRDV*VLOW
      VAR=VLOW
      GO TO 2015
C**
 2113 RSUM=ABS(SUML/SUMH)
      IF(RSUM.LT.0.15) VAR=VLOW+0.3*DIFV
      IF(RSUM.GT.7.0) VAR=VLOW+0.7*DIFV
      GO TO 2016
C
 2014 DV=FRDV*SIGN(VAR,SUMM)
 2015 VAR=VAR+DV
 2016 IF(ICAL.EQ.1) TEMTUR=VAR
      IF(TEMTUR.GT.TTLOW) GO TO 2017
      WRITE(6,2100)
 2100 FORMAT(20H *** NON-CONDENSABLE)
      TEMTUR=TTLOW
      GO TO 1611
 2017 IF(ICAL.EQ.2) PRSSUR=VAR
      IF(PRSSUR.GT.(1.05*PCRIT)) GO TO 61
      GO TO LOC,(56,164,174)
C
C     INTERNAL FUNCTION          ***** CALCAB *****
C
 3000 A=B=SUMX=0.
      DO 3001 I=NFC,NLC
      B=B+BASEB(I)*X(I)
      A=A+BASEA(I)*X(I)
 3001 SUMX=SUMX+X(I)
      A=A/SUMX/TEMTUR**1.25
      B=B/SUMX/TEMTUR
      ASQDB=A*A/B
      GO TO LOC,(4001,8001,14001,22,74 )
C
C     INTERNAL FUNCTION          ***** PCRIT *****
C
 4000 ASSIGN 4001 TO LOC
      GO TO 3000
 4001 TRASH=(4.94/ASQDB)**.6666667
      PCRIT=.0867/(TRASH*B)
      GO TO     (63,73),LOS
C
C     INTERNAL FUNCTION          ***** ZFAC *****
C
C
C     ORIGINAL NEWTON-RAPHSON ITERATIVE SOLUTION FOR REDLICH-KWONG
C     HAS BEEN REPLACED WITH ANALYTICAL SOLUTION FOR CUBIC IN Z
C
C     THE VAPOR COMPRESSIBILITY  ZFCTOR IS ALWAYS THE FIRST ROOT-ZZ
C
 5000 BP=B*PRSSUR
      ZB1=-1.
      ZB2=BP*(ASQDB-1.0-BP)
      ZB3=-ASQDB*BP*BP
      ZB1OV3=ZB1/3.0
      ZALF=ZB2-ZB1*ZB1OV3
      ZBET=2.0*ZB1OV3**3-ZB2*ZB1OV3+ZB3
      ZBETOV2=ZBET/2.
      ZALFOV3=ZALF/3.
      ZCUAOV3=ZALFOV3**3
      ZSQBOV2=ZBETOV2**2
      ZDEL=ZSQBOV2+ZCUAOV3
C     FOR ZDEL +VE THERE IS ONLY ONE REAL ROOT
```

```
C     FOR ZDEL -VE THERE ARE THREE REAL ROOTS
      IF(ZDEL)5003,5003,5004
 5004 ZEPS=SQRT(ZDEL)
      ZRCU=-ZBETOV2+ZEPS
      ZSCU=-ZBETOV2-ZEPS
      ZSIR=1.0
      ZSIS=1.0
      IF(ZRCU)5007,5008,5008
 5007 ZSIR=-1.0
 5008 IF(ZSCU)5009,5010,5010
 5009 ZSIS=-1.0
 5010 ZZR=ZSIR*(ZSIR*ZRCU)**0.33333333
      ZZS=ZSIS*(ZSIS*ZSCU)**0.33333333
      ZZ=ZZR+ZZS-ZB1OV3
      GO TO 5100
 5003 ZQUOT=ZSQBOV2/ZCUAOV3
      ZROOT=SQRT(-ZQUOT)
      ZTERM=1.0-ZROOT**2
      IF(ZBET)5011,5012,5012
 5012 ZPEI=(1.570796+ATAN(ZROOT/SQRT(ZTERM)))/3.0
      GO TO 5013
 5011 ZPEI=ATAN(SQRT(ZTERM)/ZROOT)/3.0
 5013 ZFACT=2.0*SQRT(-ZALFOV3)
      ZZ=ZFACT*COS(ZPEI)-ZB1OV3
 5100 ZFCTOR=ZZ
      ZFAC=ZFCTOR
      H=BP/ZFAC
      GO TO LOC,(23,8002,14002,75)
C
C
C     INTERNAL FUNCTION        ***** LIQDEN *****
C
C... YEN AND WOODS CORRELATION ...
 6000 PST=PSV=ZCE=0.
      DO 6101 I=NFC,NLC
      PST=PST+X(I)*ATC(I)/TMOLE
      PSV=PSV+X(I)*AVC(I)/TMOLE
 6101 ZCE=ZCE+X(I)*ZCD(I)/TMOLE
      PSP=(ZCE*10.73E0*PST)/PSV
      ACQN=DPOLY(17.4425E0,-214.578E0,989.625E0,-1522.06E0,ZCE)
      IF(ZCE.GT..26E0) GO TO 6111
      BCQN=DPOLY(-3.28257E0,13.6377E0,107.4844E0,-384.211E0,ZCE)
      GO TO 6120
 6111 BCQN=DPOLY(60.2091E0,-402.063E0,501.E0,641.E0,ZCE)
 6120 DCQN=.93E0-BCQN
      TROD=TEMTUR/PST
      ARED=1.0E0-TROD
      IF(TROD.GE.1.E0) ARED=0.0E0
      SRED1=ARED**(1./3.)
      SRED2=SRED1*SRED1
      SRED4=SRED2*SRED2
      RHORS=DADDY(1E0,ACQN,BCQN,0E0,DCQN)
      E27=DADDY(.714E0,-1.626E0,-.646E0,3.699E0,-2.198E0)
      IF(TROD.GE.1.0E0) GO TO 6300
      F27=-ALOG(TROD)
      F27=.268E0*(TROD**2.0967)/(1.0E0+.8E0*(F27**.441))
      G27=.05E0+4.221E0*((1.01E0-TROD)**.75)*EXP(-7.848E0*(1.01E0-TROD))
      GO TO 6301
 6300 F27=.268E0*(TROD**2.0967)
      G27=.05E0
 6301 H27=DADDY(-10.6E0,45.22E0,-103.79E0,114.44E0,-47.38E0)
      IF(ZCE.LT..25E0) GO TO 6102
      IF(ZCE.GT..30E0) GO TO 6104
      DELPZ=(ZCE-.25E0)/.012E0
      AFAC=3.1E0+DPOLY(-.21417E-1,-.133624E0,.0619168E0,-.010875E0,
     1   DELPZ)*DELPZ
      GO TO 6103
 6104 AFAC=1.8E0
      GO TO 6103
 6102 IF(ZCE.LT..23E0) GO TO 6105
      DELPZ=(ZCE-.23E0)/.005E0
      AFAC=3.15E0+DPOLY(-.283392E-2,.358331E-2,-.31658E-2,.416557E-3,
     1   DELPZ)*DELPZ
      GO TO 6103
```

```
 6105 AFAC=3.15E0
 6103 PRS=EXP(2.302585E0*AFAC*(1.E0-(1.E0/TROD)))
      DELP=(PRSSUR/PSP)-PRS
      TT=DELP
      IF(DELP.LT..2E0) TT=0.2E0
      DELDUM=E27+F27*ALOG(TT)+G27*EXP(H27*TT)
      IF(DELP.LT.0.2E0) DELDUM=DELDUM*(DELP/.2E0)
 6113 IF(ABS(ZCE-.27E0).GT.1.E-10) GO TO 6107
      RDZC=0.
      GO TO 6112
 6107 J=3
      IF(ZCE.GT..27E0) J=1
      IF(ZCE.LT..27E0.AND.ZCE.GT..24E0) J=2
      RI1=MADDY(FRI(1,J),FRI(2,J),FRI(3,J),FRI(4,J),FRI(5,J))
      RJ1=MADDY(FRJ(1,J),FRJ(2,J),FRJ(3,J),FRJ(4,J),FRJ(5,J))
      RK1=MPOLY(FRK(1,J),FRK(2,J),FRK(3,J),FRK(4,J),TROD)
      RL1=MADDY(FRL(1,J),FRL(2,J),FRL(3,J),FRL(4,J),FRL(5,J))
      TT=DELP
      IF(DELP.LT..20E0) TT=.20E0
      RDZC=RI1+RJ1*ALOG(TT)+RK1*EXP(RL1*TT)
      IF(DELP.LT..20E0) RDZC=RDZC*(DELP/0.2E0)
 6112 RHO=RHORS+DELDUM+RDZC
      ROCRIT=(PSP*AAMW)/(ZCE*10.73E0*PST)
      RO=ROCRIT*RHO
      ZLIQ=(PRSSUR*AAMW)/(10.73*TEMTUR*RO)
      LHC=B*PRSSUR/ZLIQ
      GO TO LOC,(25,14004)
C
C     INTERNAL FUNCTION          ***** LIQPRM *****
C
 7000 ASSIGN 7001 TO LOC
      GO TO 12000
 7001 ASSIGN 7002 TO LOC
      GO TO 11000
 7002 ASSIGN 7003 TO LOC
      GO TO  9000
 7003 GO TO      (47,79),LOS
C
C     INTERNAL FUNCTION          ***** VAPPRM *****
C
 8000 ASSIGN 8001 TO LOC
      GO TO 3000
 8001 ASSIGN 8002 TO LOC
      GO TO 5000
 8002 ASSIGN 8003 TO LOC
      GO TO 10000
 8003 ASSIGN 8004 TO LOC
      GO TO 9000
 8004 GO TO      (47,69) ,LOS
C
C     INTERNAL FUNCTION          ***** EQR *****
C
 9000 DO 9001  I=NFC,NLC
      DKV=2.302585E0*LNNU(I)+LNACT(I)-LNPHI(I)
 9001 KV(I)=EXP(DKV)
      GO TO LOC,(7003,8004,69)
C
C     INTERNAL FUNCTION          ***** GASFUG *****
C
10000 ZT=ZFCTOR -1.
      ENPHI=0.
      IF(H.GT..999E0) GO TO 10002
      ENPHI=ALOG(ZFCTOR*(1.-H))
10002 ASCON=ASQDB*ALOG(1.+H)
      BT= B*TEMTUR
      AT= A*TEMTUR**1.25 /2.
      DO 10001 I=NFC,NLC
      BTT=BASEB(I)/BT
10001 LNPHI(I)= ZT*BTT-ENPHI- ASCON*(BASEA(I)/AT-BTT)
      GO TO LOC,(76,8003)
C
C     INTERNAL FUNCTION          ***** LIQACT *****
C
11000 SUMDEL=SUMV=0.
```

```
      DO 11001 I=NFC,NLC
      AV25(I)=AVW(I)*(5.7+3.0*TEMTUR/ATC(I))
      TEM=X(I)*AV25(I)
      SUMDEL=TEM*ADEL(I) + SUMDEL
11001 SUMV = TEM + SUMV
      IF(SUMV.GT.1.E-30) SUMDEL=SUMDEL/SUMV
      DO 11002 I=NFC,NLC
11002 LNACT(I)=AV25(I)*(ADEL(I)-SUMDEL)**2/TEMTUR/1.1033
      GO TO LOC,(65,7002)
C
C     INTERNAL FUNCTION        ***** LIQFUG *****
C
12000 DO 12001  I=NFC,NLC
      TRED=TEMTUR/ATC(I)
      PRED=PRSSUR/APC(I)
      J= 3 - (2/COMPNT(I))
      IF(ABS(AOMEG(I)).LT..03.AND.J.EQ.3) J=2
      ENNU=((COEFFT(J,5)*TRED+COEFFT(J,4))*TRED+COEFFT(J,3))*TRED+
     1COEFFT(J,2)/TRED+COEFFT(J,1)+((COEFFT(J,8)*TRED+COEFFT(J,7))*TRED
     2+COEFFT(J,6))*PRED+(COEFFT(J,10)*TRED+COEFFT(J,9))*PRED*PRED-
     3ALOG(PRED)/2.302585
C THIS VARIATION SUGGESTED BY GRAYSON AND STREED.
      IF(TRED.GT.1E0) TRED=1E0
12001 LNNU(I)=ENNU+AOMEG(I)*((-3.15224*TRED*TRED+8.65808)*TRED-4.23893
     1 -1.2206/TRED-.025*(PRED-.6))
      GO TO LOC,(64,7001)
C
C     INTERNAL FUNCTION        ***** ZPH *****
C
13000 SAPH=SBET=SGAM=SDTA=0.
      DO 13001  I=NFC,NLC
      SAPH=SAPH+X(I)*APH(I)
      SBET=SBET+X(I)*BET(I)
      SGAM=SGAM+X(I)*GAM(I)
13001 SDTA=SDTA+X(I)*DTA(I)
      TOK=TRE/1.8
      HBASE=(((SDTA/4.*TOK+SGAM/3.)*TOK+SBET/2.)*TOK+SAPH)*TRE
      GO TO LOC,(14006)
C
C     INTERNAL FUNCTION        ***** GETH *****
C
14000 ASSIGN 14001 TO LOC
      GO TO 3000
14001 IF(VPFRAC .NE. 1) GO TO 14003
      ASSIGN 14002 TO LOC
      GO TO 5000
14002 HDEL=DELHVL(H,ZFCTOR) * TMOLE
      GO TO 14005
14003 IF(VPFRAC .NE. 0) GO TO 14007
      ASSIGN 14004 TO LOC
      GO TO 6000
14004 HDEL=DELHVL(LHC,ZLIQ )* TMOLE
14005 ASSIGN 14006 TO LOC
      GO TO 13000
14006 GETH=HBASE- HDEL
      GO TO 14009
14007 WRITE(6,14008)
14008 FORMAT(74H0*** ENTHALPY CANNOT BE CALCULATED, VAPOR FRACTION IS IM
     1PROPERLY SPECIFIED)
      GETH= 0.
14009 GO TO      (33,52) ,LOS
      END
```

```
      SUBROUTINE ZERO(A,N)
      DIMENSION A(N)
      DO 1 I=1,N
    1 A(I)=0.
      RETURN
      END


      SUBROUTINE ZEROI(II,N)
      DIMENSION II(N)
      DO 1 I=1,N
    1 II(I)=0
      RETURN
      END
```

```
      SUBROUTINE FLASH(PEX)
C
C***** EQUIP TYPES ADBF 10
C                  MIXR 21
C     FLASH ROUTINE - 3 ENTRIES
C                                  ADBF  - ADIABATIC FLASH
C                                  ISOF  - ISOTHERMAL FLASH
C                                  MIXR  - ADIABATIC MIXING
C     PRESSURE CHANGE ALLOWED FOR ADBF + MIXR ONLY
C     PEX=0. INDICATES OUTLET PRESSURE (PEX) IS PIN(1)
C
C     EQUIP OUTPUT VECTOR CODING -
C         1.- 2. EQUIP NO + TYPE
C         3.- 6. INLET/OUTLET STREAM NOS
C         7.     OUTLET VAPOR FRACTION (*100)
C         8.- 9. (1ST) INLET/OUTLET PRESSURES (PSIA)
C
C
C*****  COMMON DECK
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/EQUIP/EQUIP(15)
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
C*****
C
      LOGICAL FLAG,FFLAG
      DATA FLAG/.FALSE./
      REAL DUM(1),NEWDIF,KS(8),OLDKS(8),OLEQ(8),EQR(8)
      INTEGER TCNT,COUNT
C***
      ENTRY ADBF
      EQTYPE=10.
      ICAL=0
      XHIN=HIN(1)
      GO TO 100
C***
      ENTRY ISOF
      ICAL=1
      GO TO 100
C***
      ENTRY MIXR
      EQTYPE=21.
      IMIX=1
      ICAL=0
      XH=XTM=0.
      CALL ZERO(XOUT(1,1),NOCOMP)
      DO 121 J=1,NIN
      XH=XH+HIN(J)
      XTM=XTM+TMIN(J)
      DO 121 I=1,NOCOMP
  121 XOUT(I,1)=XOUT(I,1)+XIN(I,J)
      HIN(1)=XH
      XHIN=HIN(1)
      TMIN(1)=XTM
      DO 122 I=1,NOCOMP
  122 XIN(I,1)=XOUT(I,1)
      GO TO 104
  100 IMIX=0
C
  104 TCNT=1
      KCNT=0
      CALL ZERO(BPOUT,60)
      CALL ZERO(OLDKS,24)
      TSAV=TEMP=TIN(1)
      PSAV=PIN(1)
      VSAV=VFIN(1)
      BBT=BPIN(1)
      DWT=DPIN(1)
C
      PX=PEX
      IF(PX.EQ.0.) PX=PIN(1)
      IPEX=0
      IF(IMIX.EQ.1.OR.PX.NE.PIN(1)) IPEX=1
```

```
      POUT(2)=POUT(1)=PIN(1)=PX
      VFOUT(1)=1.
      VFOUT(2)=0.
      COUNT=NL=0
      DO 102 I=1,NOCOMP
      IF((XIN(I,1)/TMIN(1)).LT.0.002) GO TO 102
      COUNT=COUNT+1
      IF(NL.GT.0) GO TO 101
      NF=NL=I
      GO TO 102
  101 NL=I
  102 CONTINUE
      IF(ICAL.EQ.0) GO TO 4
C
C
    2 TOUT(2)=TOUT(1)=TT=TEMP
      TMOUT(2)=TMOUT(1)=TMIN(1)
      DO 3 I=NF,NL
    3 XOUT(I,2)=XOUT(I,1)=XIN(I,1)
      IF((TEMP-BBT).GT.0.1) GO TO 160
      V=0.
      I=2
      GO TO 162
  160 IF((DWT-TEMP).LT.0.1) GO TO 161
      IF(COUNT.EQ.1) GO TO 164
C     MAKE INITIAL K-VALUE ESTIMATES - AT FEED D.P.
      CALL DEWTP(1,AD,KS)
C     ESTIMATE VAPOR FRACTION FROM TEMPS
      V=(TEMP-BBT)/(DWT-BBT)
      GO TO 45
C
  161 V=1.
      I=1
  162 CALL ENTH(-I,HOUT(I),DUM)
      II=3-I
      TMOUT(II)=0.
      CALL ZFRO(XOUT(1,II),NOCOMP)
      GO TO 53
C
  164 CALL ENTH(1,XHIN,DUM)
      V=VFIN(1)
      II=1
      GO TO 9
C
C     MUST CALC DEW + BUBBLE POINTS OF INPUT FOR PRES CHANGE OR MIXING
C
    4 IF(IPEX.EQ.0) GO TO 6
      CALL BUBTP(1,BBT,EQR)
      IF(COUNT.GT.1) GO TO 5
      DWT=BBT
      GO TO 6
    5 CALL DEWTP(1,DWT,KS)
    6 TIN(1)=DWT
      VFIN(1)=1.
      CALL ENTH(1,HV,DUM)
      DH=25.*TMIN(1)
      IF((HIN(1)+DH).LT.HV) GO TO 14
      V=1.
      ASSIGN 9 TO NRT
      II=1
      GO TO 15
C
    9 TMOUT(II)=TMIN(1)
      DO 16 J=NF,NL
   16 XOUT(J,II)=XIN(J,1)
      VFOUT(II)=V
      HOUT(II)=XHIN
      GO TO 150
C
   14 TIN(1)=BBT
      VFIN(1)=0.
      CALL ENTH(1,HL,DUM)
      IF((HIN(1)-DH).GT.HL) GO TO 21
      V=0.
```

```
      ASSIGN 22 TO NRT
      GO TO 15
   22 II=2
      IF(NOUT.EQ.1) II=1
      GO TO 9
   21 IF(COUNT.GT.1) GO TO 28
C
C **** FOR ONE COMPONENT SYSTEM, CALCULATE V DIRECTLY
C
      TOUT(2)=TOUT(1)=DWT
      V=(HIN(1)-HL)/(HV-HL)
      IF(NOUT.GT.1) GO TO 30
      II=1
      GO TO 9
   30 TMOUT(1)=TMIN(1)*V
      TMOUT(2)=TMIN(1)-TMOUT(1)
      DO 31 I=NF,NL
      XOUT(I,1)=XIN(I,1)*V
   31 XOUT(I,2)=XIN(I,1)-XOUT(I,1)
      HOUT(1)=HIN(1)*V
      HOUT(2)=HIN(1)-HOUT(1)
      GO TO 150
C
   28 TNEG=BBT
      TPOS=DWT
      AA=ABS(HIN(1))
      FNEG=(HL-HIN(1))/AA
      FPOS=(HV-HIN(1))/AA
      TEMP=(DWT*FNEG-BBT*FPOS)/(FNEG-FPOS)
      CALL ZERO(EQR,NOCOMP)
      GO TO 2
C
   10 FLAG=.TRUE.
      DO 32 I=NF,NL
      OLDIF=EQR(I)-OLEQ(I)
      NEWDIF=KS(I)-OLDKS(I)
      IF((ABS(NEWDIF)/KS(I)).GT.1.E-4) FLAG=.FALSE.
      OLDKS(I)=KS(I)
      IF(KCNT.LE.2) GO TO 33
      IF((ABS(NEWDIF)/KS(I)).LT.1.E-4) GO TO 33
      QWEG=1.-OLDIF/NEWDIF
      IF(ABS(QWEG).LT.1.E-6) GO TO 33
      QWEG=1./QWEG
      IF(QWEG.GT..5) QWEG=.5
      KS(I)=QWEG*EQR(I)+(1.-QWEG)*KS(I)
   33 OLEQ(I)=EQR(I)
   32 EQR(I)=KS(I)
      IF(FLAG.OR.KCNT.GT.10) GO TO 48
C
      V=.5
      DO 42 COUNT=1,10
      SUM=DSUM=0.
      DO 43 I=NF,NL
      TEMPO=(KS(I)-1.)/(V*(KS(I)-1.)+1.)
      TEMPA=TEMPO*XIN(I,1)
      DSUM=DSUM+TEMPA*TEMPO
   43 SUM=SUM+TEMPA
      TEMPA=SUM/DSUM
      IF(ABS(TEMPA).LT.1.E-4) GO TO 45
      V=V+TEMPA
      IF(V.GT.1.) V=0.99
   42 IF(V.LT.0.) V=0.01
   45 VFIN(1)=V
      SUM=DSUM=0.
      DO 47 I=NF,NL
      TEMPA=(1.-V)*XIN(I,1)/(V*(KS(I)-1.)+1.)
      IF(TEMPA.LT.0.) TEMPA=0.
      TEMPO=XIN(I,1)-TEMPA
      IF(TEMPO.GT.0.) GO TO 46
      TEMPA=XIN(I,1)
      TEMPO=0.
   46 DSUM=DSUM+TEMPO
      SUM=SUM+TEMPA
      XOUT(I,2)=TEMPA
```

```
   47 XOUT(I,1)=TEMPO
      TMOUT(1)=DSUM
      TMOUT(2)=SUM
C
   11 CALL KVAL(-2,AD,KS)
      CALL KVAL(-1,AD,KS)
      KCNT=KCNT+1
      GO TO 10
C
   48 CALL ZERO(HOUT,2)
      DO 52 J=1,2
   52 IF((TMOUT(J)/TMIN(1)).GT.0.001) CALL ENTH(-J,HOUT(J),DUM)
      IF(ICAL.EQ.1) GO TO 53
      FTEMP=(HOUT(1)+HOUT(2)-HIN(1))/TMOUT(1)
      IF(ABS(FTEMP).LT.25.) GO TO 53
      IF(TCNT.LE.10) GO TO 55
      WRITE(6,56)
   56 FORMAT(* FLASH DID NOT CONVERGE*)
      GO TO 53
   55 IF(FTEMP.GT.0.) GO TO 73
      TNEG=TEMP
      FNEG=FTEMP
      GO TO 74
   73 TPOS=TEMP
      FPOS=FTEMP
C
   74 TT=(TPOS*FNEG-TNEG*FPOS)/(FNEG-FPOS)
      IF(ABS(TT-TEMP).LT.0.05) GO TO 53
      FF=ABS(FNEG/FPOS)
      DTF=TPOS-TNEG
      IF(FF.GT.10.) TT=TNEG+0.7*DTF
      IF(FF.LT.0.1) TT=TNEG+0.3*DTF
C
   60 TEMP=TT
      TCNT=TCNT+1
      TOUT(2)=TOUT(1)=TIN(1)=TT
      KCNT=0
      CALL ZERO(EQR,NOCOMP)
      GO TO 11
C
   53 IF(V.LT.0.999) GO TO 61
      II=2
      GO TO 62
   61 IF(V.GT.0.001) GO TO 64
      II=1
   62 TMOUT(II)=0.
      CALL ZERO(XOUT(1,II),NOCOMP)
   64 IF(NOUT.EQ.1) GO TO 66
      GO TO 150
C
   66 V=TMOUT(1)/(TMOUT(1)+TMOUT(2))
      XHIN=HOUT(1)+HOUT(2)
      II=1
      GO TO 9
C
C     *** INTERNAL FUNCTION GETTP
   15 TIN(1)=TSAV
      VFIN(1)=V
      CALL TSUBH(1,TEMP,DUM)
      TOUT(2)=TOUT(1)=TEMP
      GO TO NRT,(9,22)
C
  150 TIN(1)=TSAV
      PIN(1)=PSAV
      VFIN(1)=VSAV
C
C     CALCULATE BUBBLE , DEW POINTS FOR OUTLETS
C
      JJ=1
      IF(NOUT.EQ.1.OR.TMOUT(2).EQ.0.) GO TO 154
      IF(TMOUT(1).GT.0.) GO TO 151
      JJ=2
      GO TO 154
  151 DO 152 J=1,2
```

```
      CALL BUBTP(-J,BPOUT(J),KS)
      CALL DEWTP(-J,DPOUT(J),KS)
  152 CONTINUE
      GO TO 155
  154 BPOUT(JJ)=BBT
      DPOUT(JJ)=DWT
  155 IF(ICAL.EQ.1) RETURN
C**
C     SET OUTPUT PARAMETERS
C
      CALL ZERO(EQUIP,15)
      EQUIP(2)=EQTYPE
      EQUIP(7)=100.*V
      EQUIP(8)=PSAV
      EQUIP(9)=PX
      RETURN
      END
```

F EQUIPMENT ROUTINES

The equipment routines are all design-oriented, simulation-type modules, each of which computes the equipment size and cost for processing a given stream to some specified extent. With the exception of the column routine, DIST, all receive input values through the CALL argument list and return output values through the /EQUIP/ labelled COMMON vector.

DIST is the column routine. It operates in the design mode, i.e. computes column size for specified product compositions. The tray requirements are computed by the approximate pseudo-binary method proposed by Hengstebeck[3]. The method combines all components in a pair of "equivalent (multicomponent) keys" the separation of which can be computed by the McCabe-Thiele method, assuming constant mole flows of liquid and vapor in each column section. Equilibrium relations are represented by a constant key relative volatility and the McCabe-Thiele tray calculation is made with the analytical procedure described by Stoppel[4]. Column sizing (for 4" bubble caps) is achieved with the simplified method described by Bolles[5]. The costing method is based on the column shell weight and tray diameter.

HXER is the countercurrent heat exchanger design routine. It computes exchanger area and cost for exchange between two streams with specifield inlet and outlet temperatures. The calculational path depends to some extent on the stream types (service, pseudo-service or process) but the algorithm basically depends on constant film heat transfer coefficients. Each side film coefficient is valued according to the phase condition and pressure of the fluid in question as shown in the program listing. Values were obtained from Peters and Timmerhaus[6]. The overall coefficient is obtained by summing of the film resistances. The exchanger area is obtained through a ten step (in Q) numerical integration along the exchanger length. For interpolation purposes it is assumed that stream temperature is a linear function of enthalpy within each phase segment of the T vs H curve.

COMP is the single stage compressor routine which estimates the power requirement for compressing a stream to a specified pressure. The algorithm is based on a single polytropic compression coefficient, n, i.e.

$$\frac{T_{OUT}}{T_{IN}} = \left(\frac{P_{OUT}}{P_{IN}}\right)^{\frac{n-1}{n}} \qquad \text{II.1}$$

Then since the compression is assumed to be adiabatic, i.e. $Q = 0$, the compression work is given by the stream enthalpy change

$$W = \Delta H \qquad \text{II.2}$$

A factor is included to account for mechanical inefficiencies in the compressor system.

The polytropic coefficient, n, is estimated as a simple linear function of stream molecular weight. Average values were obtained from Edmister[7]. The costing is for a single stage reciprocating compressor with electric motor driver.

SPLIT splits a stream linearly into two output streams.

SPLINE, SVALUE and INTER are routines required for stream energy value estimation. SPLINE sets up a cubic spline of temperature vs value/unit enthalpy to fit a set of specified points. A separate spline is created for both hot and cold temperature regions; cooling water temperature is the changeover point. SVALUE estimates the value of a given stream between specified temperature limits. The basis of the calculation is described in Appendix I.1. INTER provides energy values at any temperature by interpolation of the splines created by SPLINE.

```
      SUBROUTINE DIST
C
C***** EQUIP TYPE 100
C
C     COLUMN DESIGN MODEL
C     BASED ON HENGSTEBECKS PSEUDO-BINARY PROCEDURE
C     COMPUTES COLUMN SIZE + COST FOR SPECIFIED KEY SPLIT
C     FOR 4IN BUBBLE CAPS
C
C     EQUIP INPUT VECTOR CODING -
C         1.- 2. EQUIP NO + TYPE
C         3.     CONDENSER CONFIGURATION FLAG - 0. TOTAL CONDENSER ,
C                1. PARTIAL CONDENSER ,
C                -1. O/H PRODUCT WITHDRAWN AS VAPOR BEFORE TOTAL CONDENSER
C         4.     REFLUX RATIO FACTOR - R/RMIN
C         5.- 6. OVERHEAD LIGHT + HEAVY KEY MOLE FRACTIONS
C         7.- 8. BOTTOM LIGHT + HEAVY KEY MOLE FRACTIONS
C         9.-10. LIGHT + HEAVY KEY COMPONENT NOS
C        11.     COLUMN PRESSURE (PSIA)
C        12.     TRAY SPACING (INS)
C
C     PARAMETERS -
C        STRES - STRESS  CARBON STEEL - PSI
C        TREFF - TRAY EFFICIENCY - FRACTION
C        ACOL,BCOL - COST COEFFS - COLUMN
C        ATR,BTR - COST COEFFS - TRAYS (SS BUBBLE TRAYS)
C        CFCOL - FACTOR  TOT CAP INV/COL CAP COST
C        AMORT - FRACTIONAL CHARGE ON CAPITAL/YR
C        CORR  - CORROSION ALLOWANCE FOR SHELL - INS
C
C     -------------------------
C     MATERIALS DATA - NOT IMPLEMENTED BY PROGRAM
C
C     STEEL      TEMP RANGE  STRESS  COST FACTOR  STRESS CORR. COST FACTOR
C     CARBON     TO     -50F 13750.       1.00                 1.00
C     NICKEL     TO    -150F 16000.       2.00                 1.72
C     STAINLESS BELOW -150F 18750.        3.5                  2.56
C     -------------------------
C
C     OUTPUTS -
C      D         1. O/H PRODUCT - LIQUID (PCOND=0.) , VAPOR (PCOND=-1.,1.)
C      B         2. BOTTOMS PRODUCT - LIQUID
C
C      V         3. FLOW TO CONDENSER (PARTIAL OR TOTAL) - VAPOR
C      VBAR      4. FLOW TO REBOILER - LIQUID
C
C     EQUIP OUTPUT VECTOR CODING -
C         1.- 2. EQUIP NO + TYPE
C         4.     CONDENSER CONFIGURATION FLAG (PCOND)
C         5.     OUTLET TEMP FOR PARTIAL CONDENSER (DEG R)
C         6.     REFLUX RATIO
C         7.     NO OF RECTIFYING TRAYS
C         8.     NO OF STRIPPING TRAYS
C         9.     COLUMN DIAMETER (FT)
C        10.     COLUMN THICKNESS (IN)
C        11.     MATERIAL COST FACTOR
C        12.     COLUMN SHELL WEIGHT (LBS)
C        13.     CAPITAL COST ($)
C        14.     OPERATING COST ($/YR)
C        15.     TOTAL COST ($/YR)
C
C*****  COMMON DECK
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/EMI/EMI(15,10)
      COMMON/EQUIP/EQUIP(15)
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
      COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR
C*****
```

```
C
      DIMENSION DUM(1)
      EQUIVALENCE (FEED(1),XIN(1,1))
      EQUIVALENCE (TOP(1),XOUT(1,1)),(BOT(1),XOUT(1,2))
      DIMENSION ALPHA(8),DBLN(8),FEED(8),XKV(8)
      DIMENSION TOP(8),BOT(8)
      DIMENSION ROV(2),ROL(2),AREA(2),DIAM(2)
      EQUIVALENCE (XMW,DUM(1))
C
      DATA STRES,TREFF,ACOL,BCOL,ATR,BTR,CFCOL,CORR/
     1 13750.,0.7,14.5,0.7,48.,1.7,4.,0.0625/
C
      QUAD(A,B,C)=(SQRT(B**2-4.*A*C))/(2.*A)
C
C*    SET INPUT PARAMETERS
C
      PCOND=EMI(3,NE)
      RFACT=EMI(4,NE)
      YL=EMI(5,NE)
      YH=EMI(6,NE)
      XL=EMI(7,NE)
      XH=EMI(8,NE)
      NLK=EMI(9,NE)
      NHK=EMI(10,NE)
      PRES=EMI(11,NE)
      TSPACE=EMI(12,NE)
C
      PSAV=PIN(1)
      POUT(4)=POUT(3)=POUT(2)=POUT(1)=PIN(1)=PRES
      VFOUT(3)=VFOUT(1)=1.
      VFOUT(4)=VFOUT(2)=0.
      CALL ZERO(TOP,NOCOMP)
      CALL ZERO(BOT,NOCOMP)
      CALL ZERO(XOUT(1,3),NOCOMP)
      CALL ZERO(XOUT(1,4),NOCOMP)
      NL=0
      DO 4 I=1,NOCOMP
      XX=FEED(I)/TMIN(1)
      IF(XX.LT.0.002) GO TO 4
      IF(NL.GT.0) GO TO 2
      NF=NL=I
      GO TO 4
    2 NL=I
    4 CONTINUE
C     ******
C     CALCULATE Q VALUE(EQUIV. TO VAP. FRAC.) FOR ACTUAL FEED
C     USE THIS VALUE FOR THE PSEUDO-BINARY SYSTEM
C
      VFSAV=VFIN(1)
      TSAV=TIN(1)
      IF(PRES.NE.PSAV) CALL BUBTP(1,BPIN(1),TKV)
      TIN(1)=BPIN(1)
      VFIN(1)=0.
      CALL ENTH(1,HL,DUM)
      CALL DEWTP(1,DPIN(1),XKV)
      TIN(1)=DPIN(1)
      VFIN(1)=1.
      CALL ENTH(1,HV,DUM)
      VFIN(1)=VFSAV
      TIN(1)=TSAV
      QQ=(HIN(1)-HL)/(HV-HL)
C
C     CALCULATE KEY MOLE FLOWS CORRESPONDING TO GIVEN MOLE FRACTIONS
      B=(FEED(NLK)-YL*FEED(NHK)/YH)/(XL-XH*YL/YH)
      D=TMIN(1)-B
      TOP(NLK)=D*YL
      TOP(NHK)=D*YH
      BOT(NLK)=B*XL
      BOT(NHK)=B*XH
C
C     COMPUTE ALPHA VALUES - CORR TO FEED AT DEW POINT
      DO 6 I=NF,NL
    6 ALPHA(I)=XKV(I)/XKV(NHK)
C
```

```
C       SET UP LINEAR LN(D/B)  VS  LN(ALPHA) RELATION
        SLOPE=(ALOG(TOP(NLK)/BOT(NLK))-ALOG(TOP(NHK)/BOT(NHK)))/(ALOG(ALPH
       1A(NLK))-ALOG(ALPHA(NHK)))
        XINT=ALOG(TOP(NHK)/BOT(NHK))-(ALOG(ALPHA(NHK)))*SLOPE
C       ******
C
C       CALCULATE LN(D/B)'S FOR ALL COMPS FROM THEIR ALPHA'S
C       CALCULATE B'S + D'S FOR ALL NON KEYS
C
        D=B=0.
        DO 8 I=NF,NL
        DBLN(I)=SLOPE*ALOG(ALPHA(I))+XINT
        BOT(I)=FEED(I)/(EXP(DBLN(I))+1.)
        TOP(I)=FEED(I)-BOT(I)
        D=D+TOP(I)
      8 B=B+BOT(I)
        TMOUT(1)=D
        TMOUT(2)=B
C
C       CALCULATE CRITICAL LN(D/B)'S
        DBCL=DBLN(NLK)+.7*(DBLN(NLK)-DBLN(NHK))
        DBCH=DBLN(NHK)-.7*(DBLN(NLK)-DBLN(NHK))
C
C*      CALCULATE KEY PORTIONS FOR LIGHT NON KEYS + SUM FOR EFFECTIVE LK
C
        BINFL=FEED(NLK)
        BINDL=TOP(NLK)
        BINBL=BOT(NLK)
        II=NLK-1
        IF(II.LT.NF) GO TO 15
        DO 14 I=NF,II
        IF(DBLN(I)-DBCL)10,10,11
C       10 - IF LN(D/B) IS LESS THAN CRIT VALUE INCLUDE WHOLLY IN LK
C       11 - IF GREATER CALCULATE PORTION TO BE INCLUDED
     10 A1=FEED(I)
        A2=TOP(I)
        A3=BOT(I)
        GO TO 12
     11 A2=BOT(I)*EXP(DBCL)
        A3=0.
        A1=A2
     12 BINFL=BINFL+A1
        BINDL=BINDL+A2
     14 BINBL=BINBL+A3
C
C       REPEAT FOR HEAVY NON KEYS SIMILARLY
C
     15 BINFH=FEED(NHK)
        BINDH=TOP(NHK)
        BINBH=BOT(NHK)
        II=NHK+1
        IF(II.GT.NL) GO TO 21
        DO 20 I=II,NF
        IF(DBCH-DBLN(I))17,17,18
C       17 - IF LN(D/B) IS GREATER THAN CRIT VALUE INCLUDE WHOLLY IN HK
C       18 - IF LESS CALCULATE PORTION TO BE INCLUDED
     17 A1=FEED(I)
        A2=TOP(I)
        A3=BOT(I)
        GO TO 19
     18 A2=0.
        A3=TOP(I)*EXP(-DBCH)
        A1=A3
     19 BINFH=BINFH+A1
        BINDH=BINDH+A2
     20 BINBH=BINBH+A3
C       ******
C       CALCULATE PARAMETERS FOR EFFECTIVE BINARY
C
     21 XF=BINFL/(BINFL+BINFH)
        XD=BINDL/(BINDL+BINDH)
        XW=BINBL/(BINBL+BINBH)
        R1=ALOG(BINDL/BINBL)
        R2=ALOG(BINDH/BINBH)
```

```
      RV1=EXP((R1-XINT)/SLOPE)
      RV2=EXP((R2-XINT)/SLOPE)
      RV=RV1/RV2
C     RV IS THE NORMALISED REL. VOL. OF THE EFFECTIVE LK
      Q1=1.-QQ
C     ******
C*    NOW APPLY STOPPELS CALCULATION FOR THEORETICAL STAGES
C
C     COMPUTE MINIMUM REFLUX RATIO - FIND FEED/EQUIL. INTERSECTION
C
  104 SS=-Q1/QQ
      XNT=XF/QQ
      ASSIGN 105 TO KINT
      GO TO 96
  105 XX1=AA-QU
      XX2=AA+QU
      XMIN=XX1
      IF(XX1.LT.0..OR.XX1.GT.1.) XMIN=XX2
      YMIN=SS*XMIN+XNT
      SSMIN=(XD-YMIN)/(XD-XMIN)
      RMIN=SSMIN/(1.-SSMIN)
      RR=RMIN*RFACT
      R1=RR+1.
      R11=1./R1
C
C     CALCULATE INTERSECTIONS
C
C     FEED LINE + RECT OL
      XQ=(XF/QQ-XD/R1)/(RR/R1+Q1/QQ)
      YQ=XQ*RR*R11+XD*R11
C
C     EQUIL LINE + RECT OL
      SS=RR*R11
      XNT=R11*XD
      ASSIGN 94 TO KINT
      GO TO 96
   94 XE=AA+QU
      XO=AA-QU
      YE=SS*XE+XNT
      YO=SS*XO+XNT
C
C     EQUIL LINE + STRIP OL
      SS=(YQ-XW)/(XQ-XW)
      XNT=XW*(1.-SS)
      ASSIGN 95 TO KINT
      GO TO 96
   95 XED=AA+QU
      XOD=AA-QU
      YED=SS*XED+XNT
      YOD=SS*XOD+XNT
C
C     CALCULATE TR TS
      TR=ALOG(((XD-YO)*(XE-XQ))/((YE-XD)*(XQ-XO)))/ALOG((YO*XE)/(XO*YE))
      TS=ALOG(((YQ-YOD)*(XED-XW))/((YED-YQ)*(XW-XOD)))/ALOG((YOD*XED)/(X
     1OD*YED))
C
C     ******
C
C     SET CONDENSER + REBOILER FLOWS
      CDFACT=RR+1
      IF(PCOND.EQ.-1.) CDFACT=RR
      RBFACT=(RR*D+TMIN(1)*(1.-QQ)-B)/B
      IF(PCOND.LE.0.) GO TO 109
C
C*    COMPUTE PARTIAL CONDENSER
C     TOP  ARE VAPOR FLOWS FROM P.C.
C     COMPUTE LIQUID FLOWS FROM P.C. + VAPOR FLOWS TO P.C.
C     USE OUTPUT VECTOR 4 AS TEMP LOCATION FOR LIQUID FLOWS
C
C     COMPUTE K-VALUES FOR O/H AT DEW POINT
      CALL DEWTP(-1,DWT,XKV)
C
      TMOUT(4)=TMOUT(3)=0.
      DO 108 I=NF,NL
```

```
      XOUT(I,4)=RR*TOP(I)/XKV(I)
      TMOUT(4)=TMOUT(4)+XOUT(I,4)
      XOUT(I,3)=TOP(I)+XOUT(I,4)
  108 TMOUT(3)=TMOUT(3)+XOUT(I,3)
      TOUT(4)=DWT
      CALL DENS(-4,ROPC,DUM)
      ROPC=ROPC*XMW
      GO TO 112
C
  109 TMOUT(3)=D*CDFACT
      DO 111 I=NF,NL
  111 XOUT(I,3)=CDFACT*TOP(I)
  112 TMOUT(4)=RBFACT*B
      DO 113 I=NF,NL
  113 XOUT(I,4)=RBFACT*BOT(I)
C
C     SUBTRACT 1 TRAY FOR PARTIAL CONDENSER
      IF(PCOND.EQ.1.) TR=TR-1.
C
C     ROUND TRAY NUMBERS
      TR=TR/TREFF
      II=TR+1.
      TR=II
      TS=TS/TREFF
      II=TS+1.
      TS=II
C
C     SET OUTPUT BUBBLE/DEW POINTS , TEMPS + ENTHALPIES
C
C
      DO 114 J=1,3
      CALL BUBTP(-J,BPOUT(J),XKV)
  114 CALL DEWTP(-J,DPOUT(J),XKV)
      BPOUT(4)=BPOUT(2)
      DPOUT(4)=DPOUT(2)
      TOUT(1)=DPOUT(1)
      IF(PCOND.EQ.0) TOUT(1)=BPOUT(1)
      TOUT(3)=DPOUT(3)
      TOUT(4)=TOUT(2)=BPOUT(2)
      IF(PCOND.EQ.0.) VFOUT(1)=0.
      DO 116 J=1,3
  116 CALL ENTH(-J,HOUT(J),DUM)
      HOUT(4)=HOUT(2)*RBFACT
C
C*    CALCULATE COLUMN DIAMETER FOR 4IN BUBBLE CAPS
C     CHECK TOP + BOTTOM
C     COMPUTE DENSITIES + FLOWS
C
      DO 122 I=1,2
      J=I+2
      VFSAV=VFOUT(J)
      IF(PCOND.EQ.-1..AND.J.EQ.3) VFLOW=VFLOW*((RR+1.)/RR)
      VFOUT(J)=1.
      CALL DENS(-J,RO,DUM)
      VFLOW=TMOUT(J)/RO
      ROV(I)=XMW*RO
      IF(PCOND.LE.0..OR.I.EQ.2) GO TO 120
      ROL(1)=ROPC
      GO TO 121
  120 VFOUT(J)=0.
      CALL DENS(-J,RO,DUM)
      ROL(I)=XMW*RO
  121 AREA(I)=0.00155*VFLOW*SQRT(ROV(I)/(ROL(I)-ROV(I)))
      DIAM(I)=1.13*SQRT(AREA(I))
      VFOUT(J)=VFSAV
  122 CONTINUE
      DCOL=AMAX1(DIAM(1),DIAM(2))
      DINS=12.*DCOL
C
C     CALCULATE SHELL THICKNESS , WEIGHT +COSTS
  130 THICK=(PRES*DINS)/(1.6*STRES-1.2*PRES)+CORR
      WT=((TR+TS+3.)*TSPACE + DINS)*3.14*DINS*THICK*0.283
      CAP=(ACOL*(WT)**BCOL + (TR+TS)*ATR*(DCOL)**BTR)*CFCOL
      CYR=CAP*AMORT
C**
```

```
C     SET OUTPUT PARAMETERS
C
      CALL ZERO(EQUIP,15)
      EQUIP(2)=100.
      EQUIP(4)=PCOND
      IF(PCOND.EQ.1.) EQUIP(5)=TOUT(1)
      EQUIP(6)=RR
      EQUIP(7)=TR
      EQUIP(8)=TS
      EQUIP(9)=DCOL
      EQUIP(10)=THICK
      EQUIP(12)=WT
      EQUIP(13)=CAP
      EQUIP(15)=CYR
      RETURN
C
C     ******
C
C     INTERNAL FUNCTION TO CALC. EQUIL. + OP. LINE INTERSECTIONS
C
   96 A1=SS*(RV-1.)
      A2=SS+XNT*(RV-1.)-RV
      QU=(SQRT(A2**2-4.*A1*XNT))/(2.*A1)
      AA=-0.5*A2/A1
      GO TO KINT,(94,95,105)
      END
```

```
      SUBROUTINE HXER(IOP,IS2,TEX,Q)
C
C*****  EQUIP TYPES (IOP=) 1,2
C
C      HEAT EXCHANGER DESIGN ROUTINE
C      COMPUTES AREA + YEARLY COST FOR 1-1 COUNTERFLOW EXCHANGER
C
C      NOMENCLATURE -
C         IOP   - OPERATION TYPE (ON FIRST INPUT)
C                 1 COOLING
C                 2 HEATING
C         IS2   - 2ND STREAM TYPE
C                 1 STEAM
C                 2 WATER
C                 3 REFRIGERANT (PROPANE OR ETHYLENE)
C                 4 PROCESS STREAM
C                 5                  STREAM 1 PSEUDO-SERVICE (PARALLEL PROC)
C                 6                  STREAM 2 PSEUDO-SERVICE (PARALLEL PROC)
C                   (IPS IS PSEUDO-SERVICE STREAM NO - IF GT 0)
C
C         TEX(1) - SPEC EXIT TEMP FOR STREAM 1
C                   (IF 0. EXCHANGE TO APPROACH)
C         (FOR IS2=4 COMPUTE TO SAT TSPEC FOR LIMITING STREAM (MIN Q))
C         TEX(2) - IS2=1,2 N.A.
C                     IS2=3 REFR EVAP TEMP
C                     IS2=4 SPEC EXIT TEMP FOR STREAM 2
C         Q     - HEAT TRANSFERRED TO 1ST STREAM
C
C      PARAMETERS -
C         HRS   - NO. OF OPERATING HRS/YR
C         AEX,BEX - EXCHANGER COST COEFFS.
C         AMORT- FRACTION OF CAPITAL CHARGED/YR
C         TWAT - COOLING WATER TEMP
C         DTW  - WATER TEMP RISE
C         CWAT - COOLING WATER COST
C         APPR - MIN TEMP APPROACH (GENERAL)
C         APRR - MIN TEMP APPROACH (REFR IS2=3,4,5)
C         ARRR - MIN TEMP APPROACH (REFR IS2=3,4,5 WITH T.LT.TRRR)
C         TS   - STEAM COND. TEMP
C         HVS  - STEAM LATENT HEAT
C         CS   - STEAM COST
C         TLOW - LOW TEMP LIMITS FOR CARBON,NICKEL STEELS
C         FMLT - COST FACTORS FOR NICKEL,STAINLESS STEELS
C         CFEX - FACTOR  TOT CAP INV/EX CAP COST
C
C      PROCESS FILM COEFFICIENTS -
C                     COND OR REB                                - XU1
C                     COOLING OR HEATING PROCESS LIQUIDS         - XU2
C                     COOLING OR HEATING PROCESS VAPORS -
C                                               (PR GT PLOW)    - XU3
C                                               (PR LT PLOW)    - XU4
C
C      EQUIP OUTPUT VECTOR CODING -
C           1.- 2. EQUIP NO + TYPE
C           3.- 6. INLET/OUTLET STREAM NOS
C           7.       EXCHANGER HEAT LOAD - + FOR HEATING 1ST STREAM (BTU/HR)
C           8.       TRANSFER AREA (FT 2)
C           9.       ENTROPY INCREASE/BTU (/DEG R *10**5)
C          10.       SERVICE OR PSEUDO-SERVICE FLOW (MOLES/HR)
C          11.       REFRIGERANT TEMP LEVEL OR PSEUDO-SERVICE OUTLET TEMP
C                    SPEC (DEG R)
C          12.       MATERIAL COST FACTOR
C          13.       CAPITAL COST ($)
C          14.       OPERATING (SERVICE) COST ($/YR)
C          15.       TOTAL COST ($/YR)
C
C
C*****  COMMON DECK
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/EQUIP/EQUIP(15)
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
      COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
```

```
     1 ARRR,TRRR
C*****
C
      DIMENSION SIDUM(4,7),SODUM(4,7)
      EQUIVALENCE (SIDUM,BPIN),(SODUM,BPOUT)
      DIMENSION TEX(2),TSPEC(2),DUM(1)
      DIMENSION HBP(2),HDP(2),TT(2),UU(2),QS(2)
      DIMENSION XTI(2),XTO(2),XHI(2),XHO(2)
      DIMENSION JVF(2),KVF(2)
      DIMENSION TLOW(2),FMLT(2)
C
      DATA AEX1,AEX2,BEX1,BEX2,CFEX/82.,25.,0.6,0.8,4./
      DATA XU1,XU2,XU3,XU4/250.,150.,80.,40./
      DATA PLOW/50./
      DATA TLOW,FMLT/410.,310.,2.,3.5/
C
      WRITE(6,600)TEX,IOP,IS2
  600 FORMAT(//3H **,* EXCHANGER -  TSPECS*,2F8.1,* IOP,IS2*,2I4/)
C
      TSPEC(1)=TEX(1)
      TSPEC(2)=TEX(2)
      SIGNQ=(3-2*IOP)
      IPS=IS2-4
      FM=1.
      AP=APPP
      IF(IS2.GT.2) AP=APRR
      IF(TSPEC(1).GT.0..AND.TSPEC(1).LT.TRRR) AP=ARRR
C
C     SET TEMPS ETC FOR IS2=1,2,3
C
      GO TO (15,20,25,100,100,100)IS2
   15 TIN(2)=TS
      HPM=18.*HVS
      DT2=0.
      U2=500.
      CST=CS
      GO TO 30
   20 TIN(2)=TWAT
      HPM=18.*DTW
      DT2=DTW
      CST=CWAT
      U2=250.
      GO TO 30
   25 TIN(2)=TSPEC(2)
      DT2=0.
      U2=250.
   30 TOUT(2)=TIN(2)+DT2
C
C**   FIND LIMITING STREAM (MIN Q)
  100 IFL=1
      DO 124 J=1,2
      IF(J.EQ.2.AND.IS2.LT.4) GO TO 200
      IF(TSPEC(J).EQ.0.) TSPEC(J)=TIN(3-J)+AP*SIGNQ*FLOAT(3-2*J)
      TT=TSPEC(J)
      GO TO 150
  122 QS(J)=DH
  124 Q=QS(1)
C*
      QR=ABS(QS(1)/QS(2))
      IF(ABS(QR-1.).LT.0.02) GO TO 200
      IF(QR.LT.1.) GO TO 130
C     2ND STREAM LIMITING
      J=1
      GO TO 132
C     FIRST STREAM LIMITING
  130 J=2
C     COMPUTE Q + NON LIMITING STREAM CONDITION
  132 ISIGN=3-2*J
      SIGN=ISIGN
      Q=-SIGN*QS(J+ISIGN)
      IF(IPS.EQ.1) QR=1./QR
      IF((3-J).EQ.IPS) Q=Q*QR
      IF(IPS.GT.0) GO TO 200
      HH=HIN(J)+SIGN*Q
```

```
      WRITE(6,625)J,ISIGN,QS,Q,HIN(J),HH
  625 FORMAT(* J,ISIGN,QS,Q,HIJ,HOJ*,2I5,2F10.0,5X,F10.0,5X,2F10.0)
      IFL=2
      GO TO 150
C
C*    ROUTINE TO CALC ENTHALPY CHANGE FOR STREAM J (HOJ-HIJ)
C     OR COMPUTE OUTLET STREAM J CONDITION FOR GIVEN ENTHALPY
C
C     SAVE INPUT 1 IN INPUT 3
C     TRANSFER INPUT J TO INPUT 1
  150 III=1
      IIJ=3
      DO 166 II=1,2
      IF(III.EQ.IIJ) GO TO 166
      DO 162 K=1,7
  162 SIDUM(IIJ,K)=SIDUM(III,K)
      DO 164 I=1,NOCOMP
  164 XIN(I,IIJ)=XIN(I,III)
      III=J
      IIJ=1
  166 CONTINUE
C     SAVE OUTPUT (3-J) IN OUTPUT 3
      DO 170 K=1,7
  170 SODUM(3,K)=SODUM(3-J,K)
      DO 174 I=1,NOCOMP
  174 XOUT(I,3)=XOUT(I,3-J)
C
      NIN=NOUT=1
      IF(IFL.EQ.2) GO TO 176
      TIN(1)=TT
      CALL ISOF(0.)
      GO TO 178
  176 HIN(1)=HH
      CALL ADBF(0.)
C
C     RESTORE INPUT 1
  178 DO 180 K=1,7
  180 SIDUM(1,K)=SIDUM(3,K)
      DO 182 I=1,NOCOMP
  182 XIN(I,1)=XIN(I,3)
C     RESET OUTPUTS 1 + 2 - OUTPUT 1 INTO J , THEN 3 INTO (3-J)
      II=J
      IJ=1
  184 DO 186 K=1,7
  186 SODUM(II,K)=SODUM(IJ,K)
      DO 188 I=1,NOCOMP
  188 XOUT(I,II)=XOUT(I,IJ)
      IF(II.EQ.(3-J)) GO TO 190
      II=3-J
      IJ=3
      GO TO 184
  190 DH=HOUT(J)-HIN(J)
      IF(IFL.EQ.1) GO TO 122
C
C     *****
C**   COMPUTE EXCHANGER AREA
C
C     WORK IN DIRN OF INCR TEMP FOR STREAM 1
C     DIVIDE INTO 10 SECTIONS + INTEGRATE NUMERICALLY WITH Q
C     ASSUME T LINEAR WITH Q WITHIN EACH PHASE SEGMENT
C
C*    COMPUTE HBP + HDP FOR EACH STREAM + SET UP BOUNDARY VALUES
  200 DO 210 J=1,2
      ISV=0
      IF(J.EQ.2.AND.IS2.LT.4) ISV=1
      IF(ISV.EQ.1) GO TO 204
      VV=VFIN(J)
      DO 202 KK=1,2
      VV=2.*VV+1.
      IF(VV.GT.1.001.AND.VV.LT.2.999) VV=2.
      KVF(KK)=VV
  202 VV=VFOUT(J)
      JVF(J)=1
      IF(KVF(1).EQ.KVF(2)) GO TO 204
```

```
      JVF(J)=0
      TSAV=TIN(J)
      VFSAV=VFIN(J)
      TIN(J)=BPIN(J)
      VFIN(J)=0.
      CALL ENTH(J,HBP(J),DUM)
      TIN(J)=DPIN(J)
      VFIN(J)=1.
      CALL ENTH(J,HDP(J),DUM)
      TIN(J)=TSAV
      VFIN(J)=VFSAV
  204 CONTINUE
C     SET UP INPUTS/OUTPUTS TO GIVE INCR TEMP FOR STREAM 1
C     Q+ HEATING FIRST STREAM , - COOLING
      JJ=J
      IF(Q.LT.0.) JJ=3-J
      GO TO (206,208)JJ
  206 XTI(J)=TIN(J)
      XTO(J)=TOUT(J)
      IF(ISV.EQ.1) GO TO 210
      XHI(J)=HIN(J)
      XHO(J)=HOUT(J)
      GO TO 210
  208 XTI(J)=TOUT(J)
      XTO(J)=TIN(J)
      IF(ISV.EQ.1) GO TO 210
      XHI(J)=HOUT(J)
      XHO(J)=HIN(J)
  210 CONTINUE
C
      WRITE(6,601)Q,XTI,XTO,XHI,XHO
  601 FORMAT(* Q,XTI,XTO,XHI,XHO*,F10.0,2F8.1,5X,2F8.1,10X,2F10.0,
     1 5X,2F10.0)
C
C*    INITIALIZE + INTEGRATE
      DQ=ABS(Q)/10.
      DTO=ABS(XTI(1)-XTI(2))+0.01
      RTO=1./XTI(1)-1./XTI(2)
      DENT=AREA=0.
C
      DO 240 K=1,10
      QT=DQ*FLOAT(K)
      FRQ=QT/ABS(Q)
C
      DO 230 J=1,2
      IF(J.EQ.2.AND.IS2.LT.4) GO TO 226
      HJ=XHI(J)+QT
      IF(J.EQ.IPS) HJ=XHI(J)+QT/QR
      IF(K.LT.10) GO TO 213
      TT(J)=XTO(J)
      GO TO 215
  213 IF(JVF(J).EQ.0) GO TO 215
      AA=XTI(J)
      BB=FRQ
      CC=XTO(J)
      GO TO 219
C
  215 BP=BPIN(J)
      DP=DPIN(J)
      HB=HBP(J)
      HD=HDP(J)
C+
      IF(BPIN(J).GT.260.) GO TO 214
      BP=AMIN1(TIN(J),TOUT(J))
      HB=AMIN1(HIN(J),HOUT(J))
C+
  214 IF(K.EQ.10) GO TO 219
      IF(HJ.GT.HD) GO TO 216
      IF(HJ.LT.HB) GO TO 218
C     TWO PHASE
      AA=BP
      BB=(HJ-HB)/(HD-HB)
      CC=DP
      GO TO 220
```

```
C     VAPOR
  216 AA=DP
      BB=(HJ-HD)/(XHO(J)-HD)
      CC=XTO(J)
      GO TO 220
C     LIQUID
  218 AA=XTI(J)
      BB=(HB-HJ)/(HB-XHI(J))
      CC=BP
C
C     EVALUATE U S AT MIDPOINT
  219 IF(JVF(J).EQ.0) GO TO 220
      IVF=KVF(J)
      GO TO 221
  220 HHJ=HJ-0.5*DQ
      IF(J.EQ.IPS) HHJ=HJ-0.5*DQ/QR
      IVF=2
      IF(HHJ.GT.HD) IVF=3
      IF(HHJ.LT.HB) IVF=1
  221 GO TO (222,223,224)IVF
  222 UU(J)=XU2
      GO TO 225
  223 UU(J)=XU1
      GO TO 225
  224 UU(J)=XU3
      IF(PIN(J).LT.PLOW) UU(J)=XU4
C
  225 IF(K.EQ.10) GO TO 230
      TT(J)=AA+BB*(CC-AA)
      GO TO 230
C
  226 TT(2)=XTI(2)+FRQ*DT2
      UU(2)=U2
  230 CONTINUE
C*
      DTN=ABS(TT(1)-TT(2))
      U=1./(1./UU(1)+1./UU(2))
      RTN=1./TT(1)-1./TT(2)
C
C     CALCULATE AREA INCREMENT
      TEST=0.9*AP
      IF(DTO.LT.TEST.OR.DTN.LT.TEST) WRITE(6,730)DTO,DTN
  730 FORMAT(* WARNING - BELOW APMIN - DTO,DTN*,2F8.1)
      XLMTD=(DTO-DTN)/(ALOG(DTO/DTN))
      AREA=AREA+DQ/(U*XLMTD)
C     ENTROPY INCREASE
      DENT=DENT+ABS(DQ*0.5*(RTO+RTN))
      DTO=DTN+0.01
      RTO=RTN
  240 CONTINUE
C*****
C     CALCULATE COSTS
C     CHECK IF ANY TEMPS BELOW MATERIAL LIMITS
C
  500 TLW=AMIN1(XTI(1),XTI(2))
      IF(TLW.LT.TLOW(1)) FM=FMLT(1)
      IF(TLW.LT.TLOW(2)) FM=FMLT(2)
      AEX=AEX1
      BEX=BEX1
      IF(AREA.LT.400.) GO TO 504
      AEX=AEX2
      BEX=BEX2
  504 CAP=FM*CFEX*AEX*(AREA)**BEX
      COP=0.
      IF(IS2.GT.2) GO TO 510
      FLSRV=ABS(Q)/HPM
      COP=CST*FLSRV*HRS
  510 CYR=AMORT*CAP+COP
C**
C     SET OUTPUT PARAMETERS
C
      CALL ZERO(EQUIP,15)
      EQUIP(2)=IOP
      EQUIP(7)=Q
```

```
EQUIP(8)=AREA
EQUIP(9)=1.E5*DENT/ABS(Q)
IF(IS2.LT.3) EQUIP(10)=FLSRV
IF(IPS.GT.0) EQUIP(10)=QR*TMIN(IPS)
IF(IS2.EQ.3) EQUIP(11)=TSPEC(2)
IF(IPS.GT.0) EQUIP(11)=TSPEC(IPS)
EQUIP(12)=FM
EQUIP(13)=CAP
EQUIP(14)=COP
EQUIP(15)=CYR
RETURN
END
```

```
      SUBROUTINE COMP(PDIS)
C
C*****  EQUIP TYPE 11
C
C       COMPRESSOR DESIGN ROUTINE
C       COMPUTES POWER REQUIREMENTS,CAPITAL+POWER COSTS FOR SINGLE
C       STAGE ISENTROPIC COMPRESSION
C
C       TOUT/TIN = (POUT/PIN)**((COEFF-1.)/COEFF)
C       Q ASSUMED ZERO , THEN W=-DELTA(H)
C
C       NOMENCLATURE -
C          PDIS   - DISCHARGE PRESSURE
C       PARAMETERS -
C          COEFF  - ISENTROPIC TEMP COEFF
C          HRS    - NO. OF OPERATING HOURS/YEAR
C          EFF    - OVERALL MECHANICAL EFFICIENCY FACTOR
C          ACOMP,BCOMP - COST COEFFS. FOR COMPRESSOR
C          AMOT,BMOT - COST COEFFS. FOR COMPRESSOR MOTOR
C          CKWH   - POWER COST/KWHR
C          AMORT  - FRACTION OF CAPITAL COSTS CHARGED/YEAR
C          CFCP   - FACTOR  TOT CAP INV/COMPR CAP COST
C
C
C       EQUIP OUTPUT VECTOR CODING -
C           1.- 2. EQUIP NO + TYPE
C           3.- 6. INLET/OUTLET STREAM NOS
C           7.     COMPRESSOR HP
C           8.- 9. INLET/OUTLET PRESSURES (PSIA)
C          13.     CAPITAL COST ($)
C          14.     OPERATING (POWER) COST ($/YR)
C          15.     TOTAL COST ($/YR)
C
C
C*****  COMMON DECK
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/EQUIP/EQUIP(15)
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
      COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR
C*****
C
      DIMENSION DUM(8)
C
      DATA EFF,ACOMP,BCOMP,AMOT,BMOT,CFCP/0.9,480.,0.76,34.,1.,2.5/
      DATA CF1,CF2,NC3/1.31,0.1,5/
C
      COST(A,B)=A*HP**B
C
      DO 1 I=1,NOCOMP
    1 XOUT(I,1)=XIN(I,1)
      TMOUT(1)=TMIN(1)
      VFOUT(1)=1.
      POUT(1)=PDIS
      PR=PDIS/PIN(1)
C     COMPUTE COEFF  (= CF1 - CF2*FRACTION C3+)
      FC3=0.
      DO 2 I=NC3,NOCOMP
    2 FC3=FC3+XIN(I,1)
      FRC3=FC3/TMIN(1)
      COEFF=CF1-CF2*FRC3
C     COMPUTE DISCHARGE TEMP
      TOUT(1)=TIN(1)*PR**((COEFF-1.)/COEFF)
C
C     COMPUTE OUTLET BUBBLE + DEW POINT TEMPS
      CALL BUBTP(-1,BPOUT(1),DUM)
      CALL DEWTP(-1,DPOUT(1),DUM)
C     COMPUTE ENTHALPY CHANGE + HP (ADJUST FOR EFFICIENCY)
      CALL ENTH(-1,HOUT(1),DUM)
      HP=(HOUT(1)-HIN(1))*3.93E-4/EFF
      CPWR=0.746*HP*HRS*CKWH
C
```

```
C     CALCULATE CAPITAL + YEARLY COSTS
C
      CCOMP=COST(ACOMP,BCOMP)
      CMOT=COST(AMOT,BMOT)
      CAP=CFCP*(CCOMP+CMOT)
      CYR=CPWR+AMORT*CAP
C**
C     SET OUTPUT PARAMETERS
C
      CALL ZERO(EQUIP,15)
      EQUIP(2)=11.
      EQUIP(7)=HP
      EQUIP(8)=PIN(1)
      EQUIP(9)=PDIS
      EQUIP(13)=CAP
      EQUIP(14)=CPWR
      EQUIP(15)=CYR
      RETURN
      END
```

```
      SUBROUTINE SPLIT(FR1)
C
C*****  EQUIP TYPE 20
C
C      SPLITS INPUT LINEARLY INTO 2 OUTPUTS
C
C      EQUIP OUTPUT VECTOR CODING -
C          1.- 2. EQUIP NO + TYPE
C          3.- 6. INLET/OUTLET STREAM NOS
C          7.- 8. 1ST,2ND OUTLET STREAM FLOW FRACTIONS (*100)
C
C
C*****  COMMON DECK
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/EQUIP/EQUIP(15)
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
C*****
C
      FR2=1.-FR1
      BPOUT(2)=BPOUT(1)=BPIN(1)
      DPOUT(2)=DPOUT(1)=DPIN(1)
      POUT(2)=POUT(1)=PIN(1)
      TOUT(2)=TOUT(1)=TIN(1)
      VFOUT(2)=VFOUT(1)=VFIN(1)
      HOUT(1)=HIN(1)*FR1
      HOUT(2)=HIN(1)*FR2
      TMOUT(1)=TMIN(1)*FR1
      TMOUT(2)=TMIN(1)*FR2
      DO 1 I=1,NOCOMP
      XX=XIN(I,1)
      XOUT(I,1)=XX*FR1
    1 XOUT(I,2)=XX*FR2
C**
C      SET OUTPUT PARAMETERS
C
      CALL ZERO(EQUIP,15)
      EQUIP(2)=20.
      EQUIP(7)=100.*FR1
      EQUIP(8)=100.*FR2
      RETURN
      END
```

```
      SUBROUTINE SPLINE(JCH)
C
C     SETS UP CUBIC SPLINE FOR ENERGY VALUE $/BTU (V)  VS  TEMP DEG R (T)
C     X=TEMP , Y=VALUE , PM - MOMENTS
C     JCH=1 HOT (NH POINTS) , 2 COLD (NC POINTS) - COMMON POINT NC+1
C
      COMMON/SPLINE/NH,NC,X(10),Y(10),PM(10)
      DIMENSION Q(10),U(10)
C
      IF(JCH.EQ.2) GO TO 20
      N1=NC+1
      N2=NC+NH+1
      GO TO 30
   20 N1=1
      N2=NC+1
C
   30 SLOPE1=(Y(N1+1)-Y(N1))/(X(N1+1)-X(N1))
      SLOPEN=(Y(N2)-Y(N2-1))/(X(N2)-X(N2-1))
      H1=X(N1+1)-X(N1)
      D1=3./H1*((Y(N1+1)-Y(N1))/H1-SLOPE1)
      H1=X(N2)-X(N2-1)
      DNP=6./H1*(SLOPEN-(Y(N2)-Y(N2-1))/H1)
      Q(N1)=-0.5
      U(N1)=D1
      NF=N1+1
      NL=N2-1
      DO 4 I=NF,NL
      AA=(X(I+1)-X(I))/(X(I+1)-X(I-1))
      D=(6./(X(I+1)-X(I-1)))*((Y(I+1)-Y(I))/(X(I+1)-X(I))
     * -(Y(I)-Y(I-1))/(X(I)-X(I-1)))
      P=(1.-AA)*Q(I-1)+2.
      Q(I)=-AA/P
    4 U(I)=(D-(1.-AA)*U(I-1))/P
      PNP=Q(NF)+2.
      PM(N2)=(DNP-U(NF))/PNP
      DO 6 I=N1,NL
      J=N2-(I-N1+1)
    6 PM(J)=Q(J)*PM(J+1)+U(J)
      RETURN
      END
```

```
      SUBROUTINE SVALUE(JCH,TEX,TDSP,VALUE)
C
C***** EQUIP TYPE 30
C
C     COMPUTES INPUT STREAM ENERGY VALUE/YEAR
C     VALUE/BTU AS FUNCTION OF TEMP IS OBTAINED FROM INTERPOLATING SPLINE
C     ASSUMES  Q VS T  LINEAR WITHIN EACH PHASE SEGMENT
C
C     JCH   1 HOT , 2 COLD
C     TEX   EXIT TEMP SPEC
C     TDSP   TEMP DISPLACEMENT
C
C     EQUIP OUTPUT VECTOR CODING -
C          1.- 2. EQUIP NO + TYPE
C          3.- 6. INLET/OUTLET STREAM NOS
C          7.     HEAT AVAILABLE - + REFR , - WASTE HEAT (BTU/HR)
C          8.- 9. INLET/OUTLET TEMPS (DEG R)
C         14.     OPERATING COST (CREDIT) ($/YR)
C         15.     TOTAL COST ($/YR)
C
C
C*****  COMMON DECK
      COMMON/CONTL/NE,NIN,NOUT,NOCOMP
      COMMON/EQUIP/EQUIP(15)
      COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
     1 XIN(8,4)
      COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
     1 TMOUT(4),XOUT(8,4)
      COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APRR,
     1 ARRR,TRRR
C*****
C
      DIMENSION DUM(1)
C
      TSAV=TIN(1)
      VSAV=VFIN(1)
      BP=BPIN(1)
      DP=DPIN(1)
      NIN=NOUT=1
      TDP=TDSP*FLOAT(2*JCH-3)
C
      IF(JCH.EQ.2) GO TO 10
C*    HOT
      XTOUT=TIN(1)
      XHOUT=HIN(1)
      TIN(1)=XTIN=TEX
      CALL ISOF(0.)
      XHIN=HOUT(1)
      GO TO 20
C*    COLD
   10 XTIN=TIN(1)
      XHIN=HIN(1)
      TIN(1)=XTOUT=TEX
      CALL ISOF(0.)
      XHOUT=HOUT(1)
C
   20 QT=XHOUT-XHIN
      VALUE=0.
C
C**   INTEGRATE IN DIRECTION OF INCREASING TEMP
C
C*    1 LIQUID
      IF(JCH.EQ.2.AND.VSAV.EQ.1.) GO TO 40
      IF(XTIN.GT.(BP+0.1)) GO TO 30
      ASSIGN 30 TO IVAL
      IF(XTOUT.LT.(BP-0.1)) GO TO 95
      TB=BP
      VFIN(1)=0.
      GO TO 90
C
C*    2 TWO-PHASE
   30 IF(XTIN.GT.(DP+0.1)) GO TO 40
      ASSIGN 40 TO IVAL
      IF(XTOUT.LT.(DP-0.1)) GO TO 95
      TB=DP
```

```
      VFIN(1)=1.
      GO TO 90
C
C*    3 VAPOR
   40 GO TO 95
C
   50 TIN(1)=TSAV
      VFIN(1)=VSAV
C**
C     SET OUTPUT PARAMETERS
C
      CALL ZERO(EQUIP,15)
      EQUIP(2)=30.
      EQUIP(7)=QT
      EQUIP(8)=TIN(1)
      EQUIP(9)=TEX
C     14 IS -VE (CREDIT)
      VALUE=VALUE*HRS
      EQUIP(15)=EQUIP(14)=-VALUE
      RETURN
C
C
   90 IFIN=0
      TIN(1)=TB
      CALL ENTH(1,HB,DUM)
      GO TO 100
   95 IFIN=1
  100 TA=XTIN
      HA=XHIN
      IF(IFIN.EQ.0) GO TO 102
      TB=XTOUT
      HB=XHOUT
  102 DT=TB-TA
      DH=HB-HA
      VAL=0.
      DO 105 I=1,3
      TT=TA+0.5*FLOAT(I-1)*DT+TDP
      CALL INTER(JCH,TT,V)
  105 VAL=VAL+V
      VALUE=VALUE+DH*VAL/3.
      IF(IFIN.EQ.1) GO TO 50
      XTIN=TB
      XHIN=HB
      GO TO IVAL,(30,40)
      END
```

```
      SUBROUTINE INTER(JCH,XR,YR)
C
C     COMPUTES ENERGY VALUE/BTU (YR)  AT  TEMP (XR)  FROM CUBIC SPLINE
C
C     JCH=1 HOT , 2 COLD
C
      COMMON/SPLINE/NH,NC,X(10),Y(10),PM(10)
C
      IF(JCH.EQ.2) GO TO 20
      N1=NC+1
      N2=NC+NH+1
      GO TO 30
   20 N1=1
      N2=NC+1
   30 IF(XR.LT.X(N1)) XR=X(N1)
      IF(XR.GT.X(N2)) XR=X(N2)
      DO 1 I=N1,N2
      IF(X(I).GE.XR) GO TO 2
    1 CONTINUE
    2 J=I-1
      IF(X(I).EQ.XR) GO TO 3
      H=X(J+1)-X(J)
      YR=(PM(J)/(6.*H))*(X(J+1)-XR)*(X(J+1)-XR)*(X(J+1)-XR)
     *   +(PM(J+1)/(6.*H))*(XR-X(J))*(XR-X(J))*(XR-X(J))
     *    +(Y(J)-PM(J)*H*H/6.)*(X(J+1)-XR)/H
     *    +(Y(J+1)-PM(J+1)*H*H/6.)*(XR-X(J))/H
      GO TO 4
    3 YR=Y(I)
    4 RETURN
      END
```

II.2 System Data Structures

The major system data structures can be divided into five categories:

i) Parameter

ii) Stream

iii) Equipment

iv) Stream processing path

v) Other

The data arrays are generally stored in labelled COMMON blocks. Especially for the larger stream and equipment arrays some use is made of blank COMMON in order to conserve central memory (the CDC 6400 program loader will overlay blank COMMON but not labelled COMMON). The five system data structure categories are described below. For most arrays an indication is given as to the system section(s) with which they are associated - C = COLSYS, S & B = SMATCH & BRBND, R = RUNIT

i) Parameters

Parameters which are common to many system routines are stored in the /PARAM/ labelled COMMON block. Their nomenclature is as follows:

AMORT - Amortization factor (fraction of capital investment charged/yr)

HRS - Number of plant operating hours/yr

TWAT - Cooling water temperature (°R)

DTW - Cooling water temperature rise (°R)

CWAT - Cooling water cost ($/lb mole)

TS - Steam condensation temperature (°R)

HVS - Steam enthalpy available (BTU/lb mole)

CS - Steam cost ($/lb mole)

CKWH - Electric power cost ($/KWH)

APPP - Exchanger closest temperature approach-water cooling (°R)

APRR - Exchanger closest temperature approach-process stream and refrigerant usage (°R)

ARRR - Exchanger closest temperature approach-below TRRR (°R)

TRRR - See ARRR above.

DTF() - Stream energy pricing discount (δ) parameters

ii) Stream

(a) Working vectors

The /SIN/ and /SOUT/ COMMON blocks contain stream properties working vectors. These are conveniently used for most stream manipulation within system routines. The coding is as follows:

BPIN/BPOUT	Bubble point temperature (°R)
DPIN/DPOUT	Dew point temperature (°R)
TIN/TOUT	Temperature (°R)
PIN/POUT	Pressure (psia)
HIN/HOUT	Enthalpy (BTU)
VFIN/VFOUT	Vapor fraction
TMIN/TMOUT	Total flow (lb moles/hr)
XIN/XOUT	Component flows (lb moles/hr)

(b) General stream arrays

There are in general three types of vectors for each stream in the general stream arrays. They are:

i) Stream control vectors - SMPA(C), SMCHA(S & B)

The SMCHA matrix has two sections, one each for hot and cold streams. The coding for SMCHA vectors is given below. Note that a slightly different coding is used for the SMPA vector and this is described in the listing for COLSYS.

1. Primary stream number
2. Secondary stream number (incremented by 1 for each heat exchange match)
3. Active/inactive flag - 0. Active, 1. Inactive
4. Stream type - 1. Feed, 0. Intermediate, -1. Product

   (2. High Priority - satisfy by service only)

   (-2. Pseudo-service stream)
5. Stream sub-type - 0. Load, 1. (Heat/Refrigeration) Source
6. Not used
7. Pressure specification (psia)
8. Temperature specification (°R)

ii) Stream properties vectors - SMPB(C), SMCHB(S & B), SMRB(R)

The order of coding for these vectors corresponds exactly to that for the stream working vectors described in (a) above. Transfer between the two is accomplished conveniently by the appropriate version of the stream handling utility routine, STMOV. Note that as for the SMCHA matrix, the SMCHB matrix has two sections, one each for hot and cold streams. The SMRB matrix has a separate section for each refrigerant circuit.

iii) Mole fraction vectors SMCHX(S & B), SMRX(R)

Since the compositions of streams do not change throughout stream processing path generation and refrigeration unit calculation, it is convenient to store these constant compositions as mole fractions. Then the component flows for a primary stream and its subsequent residual streams are generated when required from the appropriate mole fraction vector. This is carried out automatically by the stream handling utility routine, STMOV, during information transfer to the stream working vectors.

iii) Equipment

(a) Working vector - EQUIP

The EQUIP working vector is used primarily to output information from equipment routines. Its general coding (slightly different for the DIST routine) is as follows:

1. Equipment number

2. Equipment type number

3.-4. Inlet stream numbers

5.-6. Outlet stream numbers

7.-12. Equipment size and parameter data

13.-15. Equipment cost data

The type numbers (2.) for the presently available equipment are listed below.

1.-2. Heat exchanger (1. Cooler, 2. Heater)

10. Adiabatic (valve) expander

11. Compressor

20. Splitter

21. Mixer

30. Stream energy value module (Stream sale)

100. Distillation column

There is a special convention for inlet/outlet stream number and size for the EQUIP vector, as follows:

Hot (process) streams	+
Cold (process) streams	-
Service streams	>200 (201 steam, 202 cooling water, 203 refrigerant)
Pseudo-service streams	300 + signed stream number e.g. 301 for hot stream 1, 299 for cold stream 1

(b) Input equipment array - EMI (C)

The EMI array is used to input (column) equipment parameters to COLSYS. The coding of its vectors is described in the DIST routine listing.

(c) General equipment arrays - EMCH(S & B), EMR(R)

The coding for vectors of these arrays is identical to that for the EQUIP working vector described in (a) above. Note that the EMR array has a separate section for each refrigerant circuit.

iv) Stream processing path

The stream processing path data are stored in the /PATH/ COMMON block. The JPATH matrix stores the actual processing paths and the NPATH vector stores the number of paths (excluding the initial pre-processing path) used for each primary stream. Primary streams are each allocated a maximum of NPTHS paths (columns in the JPATH matrix) and there are separate sections for hot and cold streams. The first path for each primary stream is reserved for pressure change processing (pre-processing). Two subscript indexing functions, IDJ and IDN, are used to locate the correct positions in the JPATH and NPATH arrays for any given primary stream. The coding for the stream processing paths, stored as columns of the JPATH matrix, is given below:

1. Number of equipments in path

2.-6. Equipment numbers

7. Total path cost

8. Active (0)/inactive (>0) flag

v) Other

Other system labelled COMMON blocks are briefly described below:

/KPM/ Input process matrix to COLSYS. The coding is described in the COLSYS listing.

/PROP/ Physical properties pure component constant vectors

/CONTL/ System control information -

NE Equipment number, NIN, NOUT Number of input and output streams to an equipment, NOCOMP Maximum number of stream components.

/REFL/ Refrigerant level information

/SPLINE/ Stream energy cost spline information (Costs are in $/BTU)

/PLOPT/ Optimal process configuration information

/REFD/ Refrigeration unit input level and demand information

NAMELIST usage

Some use has been made of the FORTRAN NAMELIST free form input feature, specifically for COLSYS input and for the /PARLST/ and /COMP/ NAMELIST blocks. It has been used because of its convenience when only parts of particular arrays are to be provided and for the ease of identifying and changing system parameters.

APPENDIX III

CASE STUDY AND PROCESS DETAILS

III.1 High Pressure Case Study and Process Details

A full set of input, intermediate and final output data is given in this section for the high pressure (HP) process case. Data sets are grouped according to the four sections in which the system was run, i.e.

A Task identification

B Stream processing path generation

C Selection of optimal network configuration (Branch and bound optimization)

D Refrigeration unit.

The sequence of data within the sets is essentially the same as expected for input to and output from the programs as listed in Appendix II.1. Brief notes of explanation are provided to guide the reader through the various sections. Specific data can be identified through the system data structure descriptions given in Appendix II.2. Some comments have been interspersed in the data to further facilitate understanding. Additional title cards are identified by C*** *** and comment cards by C.. . Some blank cards have also been added.

Note that as the component physical properties data set is common to all four sections it has been removed from all but the first.

A TASK IDENTIFICATION (COLSYS Section)

i) Input (to MAINC)

The first section of data is that for component physical properties for the 7 pure components used in the study. There are 15 constants per component.

This is followed by the column system data which is provided in 4 NAMELIST groups as follows:

a) PARLST

This is the group of common system parameters.

b) KPMLST

This is the process matrix data which defines the column system configuration, coded as sets of equipment number, equipment type number and inlet and outlet stream numbers (+ inlet, - outlet).

c) SMPLST

This is the relevant stream control (SMPA) and properties (SMPB) information for process streams.

d) EMILST

This is the (column) equipment parameter information.

ii) Output (from COLSYS)

The output equipment vectors for the columns are shown on page 226. Their coding is described in the DIST routine listing.

The stream demand and properties output from COLSYS is described as input to the following system section, B.

*** HIGH PRESSURE PROCESS CASE ***

```
C*** HP  COLSYS SECTION INPUT ***

C.. COMPONENT PHYSICAL PROPERTIES DATA
 $COMP
 NOCOMP=7,COMPNT=1,2,3,4,5,6,7,0,
 $END
    1 HYDROGEN
   1.87962E+02    5.98860E+01    1.04108E+00    2.01600E+00    0.
   3.25000E+00    9.55000E-01    6.95200E+00   -4.57600E-04    9.56300E-07
  -2.07900E-10    7.94771E+00    2.76232E-02    3.04529E-01    7.00000E-02
    2 METHANE
   6.73077E+02    3.43260E+02    1.59365E+00    1.60420E+01    0.
   5.45000E+00    5.00000E+00    3.38100E+00    1.80440E-02   -4.30000E-06
   0.             3.72492E+01    4.42158E-02    2.91229E-01    2.00000E-01
    3 ETHYLENE
   7.42148E+02    5.09508E+02    1.98606E+00    2.80520E+01    9.49000E-02
   5.80000E+00    6.88000E+00    9.44000E-01    3.73500E-02   -1.99300E-05
   4.22000E-09    5.81184E+01    5.95223E-02    2.69607E-01    3.49000E-01
    4 ETHANE
   7.08347E+02    5.49774E+02    2.37046E+00    3.00680E+01    1.06400E-01
   5.88000E+00    7.88000E+00    2.24700E+00    3.82010E-02   -1.10490E-05
   0.             6.54226E+01    6.72910E-02    2.84639E-01    3.76000E-01
    5 PROPYLENE
   6.67198E+02    6.57180E+02    2.89900E+00    4.20780E+01    1.45100E-01
   6.20000E+00    9.69000E+00    7.53000E-01    5.69100E-02   -2.91000E-05
   5.88000E-09    8.42555E+01    8.53981E-02    2.74296E-01    5.22600E-01
    6 PROPANE
   6.17379E+02    6.65946E+02    3.20332E+00    4.40940E+01    1.53800E-01
   6.00000E+00    1.03500E+01    2.41000E+00    5.71950E-02   -1.75330E-05
   0.             8.90519E+01    9.35204E-02    2.76766E-01    5.07600E-01
    7 N-BUTANE
   5.50659E+02    7.65306E+02    4.08423E+00    5.81200E+01    1.95300E-01
   6.73000E+00    1.30000E+01    4.45300E+00    7.22700E-02   -2.22140E-05
   0.             1.12195E+02    1.20496E-01    2.73879E-01    5.84700E-01
C.. COLUMN SYSTEM DATA
 HP  COLUMN SYSTEM
 $PARLST
 AMORT=0.3,HRS=8000.,
 TWAT=535.,DTW=10.,CWAT=3.6E-5,TS=825.,HVS=1.00E3,CS=1.80E-2,CKWH=0.007,
 APPP=15.,APRR=10.,ARRR=5.,TRRR=310.,
 $END
C.. COLUMN NO.    1 DEMETHANIZER
C..               2 DEETHANIZER
C..               3 C2 SPLITTER
C..               4 DEPROPANIZER
C..               5 C3 SPLITTER
 $KPMLST
 NIS=11,NKPM=5,
 KPM1=1,100,1,-2,-3,0,
 KPM2=2,100,3,-4,-5,0,
 KPM3=3,100,4,-6,-7,0,
 KPM4=4,100,5,-8,-9,0,
 KPM5=5,100,8,-10,-11,0,
 $END
 $SMPLST
 SMPA1=1.,1.,1.,565.,0.,400.,2*0.,
 SMPA2=2.,-1.,1.,215.,0.,-1.,2*0.,
 SMPA7=7.,-1.,1.,115.,2.,-1.,2*0.,
 SMPB1=2*0.,520.,115.,0.,1.,1500.,255.,495.,315.,210.,135.,45.,45.,0.,
 $END
 $EMILST
 EMI1=1.,100.,1.,1.1,0.65,0.01,0.01,0.43,2.,3.,565.,24.,3*0.,
 EMI2=2.,100.,0.,1.2,0.385,0.015,0.025,0.565,4.,5.,465.,24.,3*0.,
 EMI3=3.,100.,-1.,1.2,0.96,0.01,0.017,0.93,3.,4.,215.,18.,3*0.,
 EMI4=4.,100.,0.,1.2,0.25,0.035,0.04,0.835,6.,7.,200.,24.,3*0.,
 EMI5=5.,100.,0.,1.2,0.90,0.08,0.075,0.76,5.,6.,115.,18.,3*0.,
 $END
```

```
C*** HP  COLUMN EQUIPMENT VECTORS ***

    1  100      0            1.00    300.5     1.28      9      11     2.27     0.78
      1.72   13120       99361           0      29808

    2  100      0            0.00      0.0     1.05     14      13     2.87     0.81
      1.00   18694       87835           0      26351

    3  100      0           -1.00      0.0     3.49     30      26     3.44     0.47
      1.00   19039      145221           0      43566

    4  100      0            0.00      0.0     0.59      8       9     1.84     0.26
      1.00    2602       23424           0       7027

    5  100      0            0.00      0.0     6.17     29      56     3.90     0.31
      1.00   20898      226055           0      67817
```

B STREAM PROCESSING PATH GENERATION (SMATCH Section)

i) Input (to MAINS)

The title card is followed by the "features" card (1111 .) which activates (with a "1", deactivates with a "0") desired processing options in SMATCH, i.e. steam, stream sales, cooling water, refrigeration and vapor recompression.

This is followed by the system parameter NAMELIST block, PARLST.

The next card (7700) gives the numbers of hot and cold primary streams and hot and cold pseudo-service streams.

The following stream data, obtained as output from the preceding COLSYS section, is provided in two groups, one each for hot and cold streams. Within each the block of specification vectors (SMCHA) for all primary streams is first, followed by alternate stream properties vectors (SMCHB) and stream component flow vectors. The latter are immediately converted into stream mole fraction vectors (SMCHX). Note that within each (hot or cold) stream group pseudo-service streams should always precede all other streams. For this particular application there are no pseudo-services.

The final input is that for refrigerant temperature levels. These data are provided for the refrigerant level scheduling algorithm in SMATCH.

ii) Output (from SMATCH)

The output from SMATCH (stream, stream processing path and equipment data) is described as input to the following system section, C. The intermediate output produced during SMATCH execution is primarily for error detection purposes and is not shown.

```
C*** HP  SMATCH SECTION INPUT ***

 HP PLANT -PROPANE SERIES PROCESSED
    1    1    1    1    1
 $PARLST
 AMORT=0.3,HRS=8000.,
 TWAT=535.,DTW=10.,CWAT=3.6E-5,TS=825.,HVS=1.00E3,CS=1.80E-2,CKWH=0.007,
 APPP=10.,APRR=10.,ARRR=5.,TRRR=310.,
 $END
    7    7    0    0
C.. HOT STREAM DATA -
C.. STREAM CONTROL VECTORS , THEN ALTERNATE PROPERTIES + FLOWS VECTORS
    1                   -1    1                         0.0    555.0
    2                    1                            565.0    400.0
    3                    0                              0.0    301.5
    4                    0                              0.0    481.9
    5                    0                              0.0    416.0
    6                    0                              0.0    552.0
    7                    0                              0.0    508.2
  555.2    555.2    614.4       174.4  15604605    1.000    2368.7
    0.00      0.00       0.00      0.00      0.00  22214.40        0.00
  240.0    449.1    520.0       115.0   5185628    1.000    1500.0
  255.00    495.00     315.00    210.00    135.00     45.00       45.00
  240.0    348.5    348.5       565.0   1871954    1.000    1420.0
  255.80   1041.20     122.20      0.80       .00       .00         .00
  481.9    490.9    490.9       465.0   2140741    1.000    1079.7
    0.00     14.05     628.78    418.39     16.14      2.32         .00
  416.0    422.1    422.1       215.0   1794653    1.000    1091.0
    0.00     23.90    1056.26     10.83       .00      0.00        0.00
  552.0    559.7    559.7       200.0   1305193    1.000     282.3
    0.00      0.00       0.00      9.01    196.85     66.71        9.70
  508.2    512.5    512.5       115.0   3802021    1.000     987.8
    0.00      0.00       0.00     40.74    868.85     78.18        0.00

C.. COLD STREAM DATA
    1                   -1    1                       215.0     -1.0
    2                    0                              0.0    577.8
    3                    0                              0.0    648.6
    4                   -1    1                       115.0     -1.0
    5                    0                              0.0    466.6
    6                    0                              0.0    653.8
    7                    0                              0.0    547.6
  240.0    300.1-   300.1       565.0    925856    1.000     751.6
  255.00    488.16       7.84       .61       .00       .00         .00
  530.0    577.8    530.0       565.0   -975563    0.000    1023.9
     .00      9.36     420.26    286.49    184.71     61.57       61.57
  634.1    648.6    634.1       465.0   1437380    0.000    1011.0
    0.00       .00       4.61     25.79    577.15    199.15      204.27
  461.3    466.6    461.3       215.0   -454183    0.000     212.2
    0.00       .00       3.73    200.61      7.86      0.00        0.00
  461.3    466.6    461.3       215.0  -2735952    0.000    1278.2
    0.00       .00      22.44   1208.43     47.33      0.00        0.00
  644.5    653.8    644.5       200.0    581610    0.000     203.8
    0.00      0.00       0.00       .01     13.90      8.37      181.55
  533.7    547.6    533.7       115.0  -1296919    0.000     959.2
    0.00      0.00       0.00       .00     72.28    741.46      145.49

C.. REFRIGERATION LEVELS -
    5
296.
345.
385.
425.
470.
```

C SELECTION OF OPTIMAL NETWORK CONFIGURATION (BRBND Section)

i) Input (to MAINB)

The title and system parameter cards are as for the previous section with one addition, the discount (δ) parameter for cold stream energy transfers (DTF(2)).

The card containing parameters for the branch and bound optimization is next.

The remainder of the data, with the exception of the final refrigerant level/cost information, is obtained as the complete punched output from SMATCH. It is divided into three sections as described below:

(a) Stream information

The first card (7 7 0 0 24 16) gives the numbers of primary, pseudo-service and total streams for the hot and cold categories. Within each category the block of mole fraction vectors (SMCHX) for primary streams is first. This is followed by the block of alternate stream control (SMCHA) and properties (SMCHB) vectors for all primary and residual streams.

(b) Stream processing paths

Within this section information is again divided into hot and cold (primary) stream categories. Within each the first card gives the number of processing paths used for each primary stream (NPATH vector). Then follow the processing path matrix (JPATH) sections for all primary streams, each preceded by the primary stream number.

(c) Equipment vectors

The number of equipments precedes the complete listing of equipment vectors. Within each vector the equipment number is first. The second entry is the equipment type number and the coding for the remainder of the vector can be found in the corresponding equipment subroutine listing. Note that at this stage cost values for equipment involving energy costs (types 1 and 30) are missing. They are to be provided as a first step in this system section by the ENERGY routine.

The final data cards contain refrigerant level and cost information. These cost figures (in $/BTU) are either estimated or obtained from previous calculation and are updated later by the refrigeration routine, RUNIT. Final convergence of the overall problem cannot be obtained until these costs are within the correct range (refer to Figure 10, section 6.1). The values shown here are those for the final computation pass.

ii) Output from ENERGY (entry ENEC)

The ENERGY routine (specifically the ENEC entry) computes all costs associated with energy transfers, i.e. refrigerant usages and stream sales. This completes the equipment costing process and allows processing path costs to be totalled and sorted into order of increasing cost for each primary stream. The completed output from this computation phase is shown on page 239. The stars (*) which indicate optimal paths for each primary stream were added after the following branch and bound optimization stage.

Note that the first processing path for each primary stream is not shown. This is the pressure-changing or pre-processing path which does not present any processing alternatives and hence is not directly required for the branch and bound optimization calculations. The pre-processing path cost has however been added into the costs for all other paths for the appropriate stream.

Note also that costs for process/process matches are divided equally between paths for the two respective streams to avoid duplication of costs.

iii) Output from BRBND

The intermediate output produced during execution of the branch and bound optimizing routine, BRBND, is not shown. After the optimization has been completed BRBND compiles a list of the numbers of the equipment which comprise the optimal plant (NEOPT vector). This is shown on page 240 together with the optimal plant number and cost. Note that since the maximum number of processing paths per stream (NPTHS), which is also the base for the plant number, is 10 then each decimal digit of the plant number is a component path sequence number.

iv) Output from ENERGY (entry ENDS)

From the NEOPT vector the ENERGY routine (specifically the ENDS entry) compiles lists of energy transfers and overall cost statistics for the optimal plant. These are shown on page 240. Note the following points.

The temperatures given in the REFD refrigeration demand array are the maximum temperatures at which refrigeration should be supplied. Dependent on the levels available the actual temperature used may be rather lower than that shown.

As the high pressure process does not use any stream as a pseudo-service the PSR array is not used.

The costs shown do not agree exactly with the values given in Table 8 (section 7.5) as the energy cost values have yet to be updated by the final pass through the refrigeration routine.

The ENDS routine also writes out the sequence of complete equipment vectors for the optimal plant. This is not shown in the line printer output in which it is produced due to page width limitations. However complete details of the optimal plant abstracted from the SMATCH output are shown on pages 241 and 242. Both stream and equipment details are shown. Note that only data for those streams and equipment which form part of the optimal plant are presented.

```
C*** HP  BRBND SECTION INPUT ***

 HP PLANT - PROPANE SERIES PROCESSED
 $PARLST
 AMORT=0.3,HRS=8000.,
 TWAT=535.,DTW=10.,CWAT=3.6E-5,TS=825.,HVS=1.00E3,CS=1.80E-2,CKWH=0.007,
 APPP=10.,APRR=10.,ARRR=5.,TRRR=310.,DTF(2)=0.,
 $END
C.. BRANCH + BOUND PARAMETER CARD
   10     3     4     2

    7     7     0     0    24    16
C.. HOT STREAM MOLE FRACTIONS
    0.0000      0.0000      0.0000      0.0000      0.0000      1.0000      0.0000
     .1700       .3300       .2100       .1400       .0900       .0300       .0300
     .1801       .7332       .0861       .0006      0.0000      0.0000      0.0000
    0.0000       .0130       .5824       .3875       .0149       .0021      0.0000
    0.0000       .0219       .9682       .0099      0.0000      0.0000      0.0000
    0.0000      0.0000      0.0000       .0319       .6973       .2363       .0344
    0.0000      0.0000      0.0000       .0412       .8796       .0791      0.0000

C.. HOT STREAM CONTROL + PROPERTIES VECTORS
       1       0       0      -1       1             0.0      0.0   555.0       1
   555.2   555.2   614.4             174.4    15604605   1.000  2214.4
       2       0       1       1       0             0.0    565.0   400.0       2
   240.0   449.1   520.0             115.0     5185628   1.000  1500.0
       3       0       0       0       0             0.0      0.0   301.5       3
   240.0   348.5   348.5             565.0     1871954   1.000  1420.0
       4       0       0       0       0             0.0      0.0   481.9       4
   481.9   490.9   490.9             465.0     2140741   1.000  1079.7
       5       0       0       0       0             0.0      0.0   416.0       5
   416.0   422.1   422.1             215.0     1794653   1.000  1091.0
       6       0       0       0       0             0.0      0.0   552.0       6
   552.0   559.7   559.7             200.0     1305193   1.000   282.3
       7       0       0       0       0             0.0      0.0   508.2       7
   508.2   512.5   512.5             115.0     3802021   1.000   987.8
       2       0       1       1       0             0.0    565.0   400.0       8
   240.0   485.8   623.4             254.9     6788513   1.000  1500.0
       2       0       1       1       0             0.0    565.0   400.0       9
   240.0   485.8   545.0             254.9     5410782   1.000  1500.0
       2       0       1       1       0             0.0    565.0   400.0      10
   240.0   522.5   653.3             565.0     7053372   1.000  1500.0
       2       0       0       1       0             0.0    565.0   400.0      11
   240.0   522.5   545.0             565.0     4981852   1.000  1500.0
       1       1       0      -1       1             0.0      0.0   555.0      12
   555.2   555.2   555.2             174.4    11172383    .896  2214.4
       5       1       1       0       0             0.0      0.0   416.0      13
   472.3   476.7   522.8             531.2     2642552   1.000  1200.1
       7       1       0       2       0             0.0      0.0   508.2      14
   508.2   512.5   508.9             115.0    -2313502    .067   987.8
       7       1       1       0       0             0.0      0.0   508.2      15
   554.3   558.1   575.7             222.4     4969671   1.000  1086.6
       2       1       0       1       0             0.0    565.0   400.0      16
   240.0   522.5   495.0             565.0     3237481    .881  1500.0
       1       2       0      -1       1             0.0      0.0   555.0      17
   555.2   555.2   555.2             174.4     4963488    .459  2214.4
       1       2       0      -1       1             0.0      0.0   555.0      18
   555.2   555.2   555.2             174.4     5083094    .467  2214.4
       2       2       0       1       0             0.0    565.0   400.0      19
   240.0   522.5   445.0             565.0      757144    .634  1500.0
       2       2       0       1       0             0.0    565.0   400.0      20
   240.0   522.5   472.7             565.0     2124174    .772  1500.0
       1       3       0      -1       1             0.0      0.0   555.0      21
   555.2   555.2   555.2             174.4    -1125802    .029  2214.4
       1       3       0      -1       1             0.0      0.0   555.0      22
   555.2   555.2   555.2             174.4    -1125802    .029  2214.4
       2       3       0       1       0             0.0    565.0   400.0      23
   240.0   522.5   434.6             565.0      256871    .584  1500.0
       2       3       0       1       0             0.0    565.0   400.0      24
   240.0   522.5   435.0             565.0      276302    .586  1500.0

C.. COLD STREAM DATA
     .3393       .6495       .0104       .0008      0.0000      0.0000      0.0000
    0.0000       .0091       .4105       .2798       .1804       .0601       .0601
    0.0000      0.0000       .0046       .0255       .5709       .1970       .2020
```

```
  0.0000      0.0000      .0176      .9454      .0370     0.0000     0.0000
  0.0000      0.0000      .0176      .9454      .0370     0.0000     0.0000
  0.0000      0.0000     0.0000      .0000      .0682      .0411      .8908
  0.0000      0.0000     0.0000     0.0000      .0754      .7730      .1517

    1        0     1      -1       1              0.0    215.0     -1.0       1
 240.0    300.1     300.1                 565.0     925856   1.000    751.6
    2        0     0       0       0              0.0      0.0    577.8       2
 530.0    577.8     530.0                 565.0    -975563   0.000   1023.9
    3        0     0       0       0              0.0      0.0    648.6       3
 634.1    648.6     634.1                 465.0    1437380   0.000   1011.0
    4        0     1      -1       1              0.0    115.0     -1.0       4
 461.3    466.6     461.3                 215.0    -454183   0.000    212.2
    5        0     0       0       0              0.0      0.0    466.6       5
 461.3    466.6     461.3                 215.0   -2735952   0.000   1278.2
    6        0     0       0       0              0.0      0.0    653.8       6
 644.5    653.8     644.5                 200.0     581610   0.000    203.8
    7        0     0       0       0              0.0      0.0    547.6       7
 533.7    547.6     533.7                 115.0   -1296919   0.000    959.2
    1        0     0      -1       1              0.0    215.0     -1.0       8
 240.0    240.0     248.6                 215.0     925856   1.000    751.6
    4        0     0      -1       1              0.0    115.0     -1.0       9
 424.2    430.1     424.6                 115.0    -442874    .164    212.2
    5        1     0       2       0              0.0      0.0    466.6      10
 461.3    466.6     463.4                 215.0    1211037    .643   1278.2
    5        1     0       2       0              0.0      0.0    466.6      11
 461.3    466.6     464.4                 215.0    2155206    .794   1278.2
    2        1     0       2       0              0.0      0.0    577.8      12
 530.0    577.8     545.6                 565.0     729706    .415   1023.9
    7        1     0       2       0              0.0      0.0    547.6      13
 533.7    547.6     535.6                 115.0     415805    .264    959.2
    7        1     0       2       0              0.0      0.0    547.6      14
 533.7    547.6     545.2                 115.0    4792371    .913    959.2
    7        1     0       2       0              0.0      0.0    547.6      15
 533.7    547.6     545.2                 115.0    4792371    .913    959.2
    4        1     0      -1       1              0.0    115.0     -1.0      16
 424.2    430.1     426.0                 115.0      59746    .598    212.2

C.. STREAM PROCESSING PATHS
C.. HOT STREAMS -
    6    3    1    2    2    3    3
 1
    0    0    0    0    0    0
    1   14    0    0    0    0
    2   15   34    0    0    0
    3   15   35   41    0    0
    3   15   36   43    0    0
    4   15   35   42   50    0
    4   15   36   44   51    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
 2
    4    1    2    3    4    0
    3   29   38   45    0    0
    4   29   39   47   53    0
    4   29   38   46   52    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
 3
    1   16    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
    0    0    0    0    0    0
```

```
     0     0     0     0     0     0
  4
     1    17     0     0     0     0
     1    18     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
  5
     1    19     0     0     0     0
     2    20    21     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
  6
     1    22     0     0     0     0
     1    23     0     0     0     0
     1    24     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
  7
     1    25     0     0     0     0
     2    26    37     0     0     0
     2    27    28     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
C.. COLD STREAMS -
     1     3     1     3     6     1     5
  1
     1     5     0     0     0     0
     1    12     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
  2
     1     7     0     0     0     0
     1    15     0     0     0     0
     2    23    32     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
  3
     1     8     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
```

```
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
 4
     1     6     0     0     0     0
     1    13     0     0     0     0
     1    39     0     0     0     0
     2    46    49     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
 5
     1     9     0     0     0     0
     2    18    30     0     0     0
     2    21    31     0     0     0
     1    26     0     0     0     0
     1    35     0     0     0     0
     1    44     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
 6
     1    10     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
 7
     1    11     0     0     0     0
     2    24    33     0     0     0
     1    28     0     0     0     0
     2    36    40     0     0     0
     2    42    48     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
     0     0     0     0     0     0
C.. EQUIPMENT VECTORS
   53
   1   11        2     0        8      0               700
 115.0       254.9        0.0          0.0       0.0   233843      29240      99393

   2    1        8   202        9      0           -1377731
 684.5        13.7     7654.1          0.0       1.0    18549       2204       7769

   3   11        9     0       10      0               717
 254.9       565.0        0.0          0.0       0.0   238590      29964     101541

   4    1       10   202       11      0           -2071520
 828.1        17.8    11508.4          0.0       1.0    21601       3314       9795

   5   10       -1     0       -8      0               100
 565.0       215.0        0.0          0.0       0.0        0          0          0

   6   10       -4     0       -9      0                16
 215.0       115.0        0.0          0.0       0.0        0          0          0

   7    2       -2   201        0      0            4432222
  98.4        59.4      246.2          0.0       1.0     5147      35458      37002
```

8	2	-3	201	0	0		4493258		
146.9		34.7	249.6		0.0	1.0	6547	35946	37910
9	2	-5	201	0	0		6208896		
103.2		94.3	344.9		0.0	1.0	5297	49671	51260
10	2	-6	201	0	0		1377391		
47.0		32.8	76.5		0.0	1.0	3305	11019	12011
11	2	-7	201	0	0		6715581		
141.7		63.8	373.1		0.0	1.0	6408	53725	55647
12	30	-8	0	0	0		0		
249.0		-1.0	0.0		0.0	0.0	0	0	0
13	30	-9	0	0	0		0		
425.0		-1.0	0.0		0.0	0.0	0	0	0
14	1	1	202	0	0		-17157662		
8764.9		6.7	95320.3		0.0	1.0	142625	27452	70240
15	1	1	-2	12	0		-4432222		
3239.9		6.6	0.0		0.0	1.0	64331	0	19299
16	1	3	203	0	0		0		
0.0		0.0	301.5		0.0	0.0	0	0	0
17	1	4	203	0	0		0		
0.0		0.0	481.9		0.0	0.0	0	0	0
18	1	4	-5	0	-10		-3926923		
1345.2		10.4	0.0		0.0	1.0	31844	0	9553
19	1	5	203	0	0		0		
0.0		0.0	416.0		0.0	0.0	0	0	0
20	11	5	0	13	0		292		
215.0		531.2	0.0		0.0	0.0	114499	12194	46543
21	1	13	-5	0	-11		-4865113		
3192.2		7.3	0.0		0.0	1.0	63573	0	19072
22	1	6	202	0	0		-1703878		
861.5		5.3	9466.0		0.0	1.0	22295	2726	9415
23	1	6	-2	0	-12		-1703878		
836.3		5.6	0.0		0.0	1.0	21772	0	6532
24	1	6	-7	0	-13		-1703878		
668.9		6.9	0.0		0.0	1.0	18209	0	5463
25	1	7	203	0	0		0		
0.0		0.0	508.2		0.0	0.0	0	0	0
26	1	7	-5	14	0		-6208896		
1066.1		19.7	0.0		0.0	1.0	26439	0	7932
27	11	7	0	15	0		344		
115.0		222.4	0.0		0.0	0.0	130811	14365	53608
28	1	15	-7	0	0		-6628207		
3598.3		5.5	0.0		0.0	1.0	69965	0	20989
29	1	11	203	16	0		0		
0.0		0.0	495.0		0.0	0.0	0	0	0
30	2	-10	201	0	0		2261907		
37.8		93.6	125.7		0.0	1.0	2898	18095	18965
31	2	-11	201	0	0		1317738		
22.0		93.4	73.2		0.0	1.0	2097	10542	11171
32	2	-12	201	0	0		2726953		
62.5		56.5	151.5		0.0	1.0	3921	21816	22992

33	2	-13	201	0	0	5002857			
106.2		63.2	277.9		0.0	1.0	5390	40023	41640
34	1	12	202	0	0	-12725440			
6958.8		5.1	70696.9		0.0	1.0	118584	20361	55936
35	1	12	-5	17	0	-6208896			
726.0		35.4	0.0		0.0	1.0	19442	0	5833
36	1	12	-7	18	-14	-6089290			
4509.1		5.1	0.0		0.0	1.0	83806	0	25142
37	1	14	203	0	0	0			
0.0		0.0	508.2		0.0	0.0	0	0	0
38	1	16	203	19	0	0			
0.0		0.0	445.0		0.0	0.0	0	0	0
39	1	16	-9	20	0	-1125983			
224.0		25.0	0.0		0.0	1.0	8434	0	2530
40	2	-14	201	0	0	626291			
13.5		61.6	34.8		0.0	1.0	1564	5010	5480
41	1	17	202	0	0	-6516545			
3563.5		5.1	36203.0		0.0	1.0	69423	10426	31253
42	1	17	-7	21	-15	-6089290			
4509.1		5.1	0.0		0.0	1.0	83806	0	25142
43	1	18	202	0	0	-6636151			
3628.9		5.1	36867.5		0.0	1.0	70440	10618	31750
44	1	18	-5	22	0	-6208896			
726.0		35.4	0.0		0.0	1.0	19442	0	5833
45	1	19	203	0	0	0			
0.0		0.0	400.0		0.0	0.0	0	0	0
46	1	19	-9	23	-16	-500273			
285.8		7.7	0.0		0.0	1.0	9761	0	2928
47	1	20	203	24	0	0			
0.0		0.0	435.0		0.0	0.0	0	0	0
48	2	-15	201	0	0	626291			
13.5		61.6	34.8		0.0	1.0	1564	5010	5480
49	30	-16	0	0	0	426			
-1.0		0.0	0.0		0.0	0.0	0	0	0
50	1	21	202	0	0	-427255			
233.6		5.1	2373.6		0.0	1.0	8650	684	3279
51	1	22	202	0	0	-427255			
233.6		5.1	2373.6		0.0	1.0	8650	684	3279
52	1	23	203	0	0	0			
0.0		0.0	400.0		0.0	0.0	0	0	0
53	1	24	203	0	0	0			
0.0		0.0	400.0		0.0	0.0	0	0	0

C.. REFRIGERATION LEVELS + COSTS -
5

296.	.0000138
345.	.0000120
385.	.0000095
425.	.0000043
470.	.0000025

C*** HP (SORTED) STREAM PROCESSING PATH ARRAY ***

C.. * DENOTES OPTIMUM PROCESSING SEQUENCE

1								
2	4	15	36	44	51	0	28416	
3	4	15	35	42	50	0	28416	
4	3	15	35	41	0	0	43819	
5	3	15	36	43	0	0	53970	
6	2	15	34	0	0	0	65585	*
7	1	14	0	0	0	0	70240	
2								
12	4	29	39	47	53	0	456936	*
13	4	29	38	46	52	0	476838	
14	3	29	38	45	0	0	514500	
3								
22	1	16	0	0	0	0	289046	*
4								
32	1	18	0	0	0	0	4776	
33	1	17	0	0	0	0	91484	*
5								
42	2	20	21	0	0	0	56079	*
43	1	19	0	0	0	0	416059	
6								
52	1	24	0	0	0	0	2731	
53	1	23	0	0	0	0	3266	
54	1	22	0	0	0	0	9415	*
7								
62	2	26	37	0	0	0	20284	
63		27	28	0	0	0	64102	*
64	1	25	0	0	0	0	233312	
1								
72	1	12	0	0	0	0	-94110	*
2								
82	1	15	0	0	0	0	9649	*
83	2	23	32	0	0	0	26258	
84	1	7	0	0	0	0	37002	
3								
92	1	8	0	0	0	0	37910	*
4								
102	1	13	0	0	0	0	-37260	
103	2	46	49	0	0	0	-18871	
104	1	39	0	0	0	0	1265	*
5								
112	1	44	0	0	0	0	2916	
113	1	35	0	0	0	0	2916	
115	2	21	31	0	0	0	20707	*
116	2	18	30	0	0	0	23714	
117	1	9	0	0	0	0	51260	
6								
122	1	10	0	0	0	0	12011	*
7								
132	1	28	0	0	0	0	10494	*
133	2	42	48	0	0	0	18051	
134	2	36	40	0	0	0	18051	
135	2	24	33	0	0	0	44371	
136	1	11	0	0	0	0	55647	

```
C*** HP  OPTIMAL PLANT SUMMARY ***

 OPTIMAL PLANT NO.,COST        51121321113311    1030575

 EQUIPMENT NOS. -
   1  2  3  4  5  6  8 10 12 15 16 17 20 21 22 27 28 29 31 34 39 47 53


C.. REFRIGERATION DEMAND , STREAM SALE + PSEUDO-SERVICE USAGE  ARRAYS
 REFD,REFS,PRSV ARRAYS -
   296     2394842
   390     1521120
   435     1854210
   472     3926780
   485     1744371

   249          -8


 TOTAL PLANT COST SUMMARY -

 TOTAL                                                   1030575
 PROCESS NETWORK (EXCL. REFR. DEMANDS + SALES)            558506
 PRE-PROCESSING (PRESSURE SPECS)                          218498
 POST-PROCESSING (TEMPERATURE SPECS)                      340008
 REFRIGERATION DEMANDS                                    566180
 STREAM SALES                                             -94111
```

```
C*** HP  OPTIMAL PLANT DETAILS ***

C.. HOT STREAM DATA
    0.0000     0.0000     0.0000     0.0000     0.0000     1.0000     0.0000
     .1700      .3300      .2100      .1400      .0900      .0300      .0300
     .1801      .7332      .0861      .0006     0.0000     0.0000     0.0000
    0.0000      .0130      .5824      .3875      .0149      .0021     0.0000
    0.0000      .0219      .9682      .0099     0.0000     0.0000     0.0000
    0.0000     0.0000     0.0000      .0319      .6973      .2363      .0344
    0.0000     0.0000     0.0000      .0412      .8796      .0791     0.0000

     1       0       0      -1       1            0.0       0.0    555.0      1
   555.2   555.2   614.4            174.4   15604605   1.000   2214.4
     2       0       1       1       0            0.0    565.0    400.0      2
   240.0   449.1   520.0            115.0    5185628   1.000   1500.0
     3       0       0       0       0            0.0       0.0    301.5      3
   240.0   348.5   348.5            565.0    1871954   1.000   1420.0
     4       0       0       0       0            0.0       0.0    481.9      4
   481.9   490.9   490.9            465.0    2140741   1.000   1079.7
     5       0       0       0       0            0.0       0.0    416.0      5
   416.0   422.1   422.1            215.0    1794653   1.000   1091.0
     6       0       0       0       0            0.0       0.0    552.0      6
   552.0   559.7   559.7            200.0    1305193   1.000    282.3
     7       0       0       0       0            0.0       0.0    508.2      7
   508.2   512.5   512.5            115.0    3802021   1.000    987.8
     2       0       1       1       0            0.0    565.0    400.0      8
   240.0   485.8   623.4            254.9    6788513   1.000   1500.0
     2       0       1       1       0            0.0    565.0    400.0      9
   240.0   485.8   545.0            254.9    5410782   1.000   1500.0
     2       0       1       1       0            0.0    565.0    400.0     10
   240.0   522.5   653.3            565.0    7053372   1.000   1500.0
     2       0       0       1       0            0.0    565.0    400.0     11
   240.0   522.5   545.0            565.0    4981852   1.000   1500.0
     1       1       0      -1       1            0.0       0.0    555.0     12
   555.2   555.2   555.2            174.4   11172383    .896   2214.4
     5       1       1       0       0            0.0       0.0    416.0     13
   472.3   476.7   522.8            531.2    2642552   1.000   1200.1
     7       1       1       0       0            0.0       0.0    508.2     15
   554.3   558.1   575.7            222.4    4969671   1.000   1086.6
     2       1       0       1       0            0.0    565.0    400.0     16
   240.0   522.5   495.0            565.0    3237481    .881   1500.0
     2       2       0       1       0            0.0    565.0    400.0     20
   240.0   522.5   472.7            565.0    2124174    .772   1500.0
     2       3       0       1       0            0.0    565.0    400.0     24
   240.0   522.5   435.0            565.0     276302    .586   1500.0

C.. COLD STREAM DATA
     .3393      .6495      .0104      .0008     0.0000     0.0000     0.0000
    0.0000      .0091      .4105      .2798      .1804      .0601      .0601
    0.0000     0.0000      .0046      .0255      .5709      .1970      .2020
    0.0000     0.0000      .0176      .9454      .0370     0.0000     0.0000
    0.0000     0.0000      .0176      .9454      .0370     0.0000     0.0000
    0.0000     0.0000     0.0000      .0000      .0682      .0411      .8908
    0.0000     0.0000     0.0000     0.0000      .0754      .7730      .1517

     1       0       1      -1       1            0.0    215.0     -1.0      1
   240.0   300.1   300.1            565.0     925856   1.000    751.6
     2       0       0       0       0            0.0       0.0    577.8      2
   530.0   577.8   530.0            565.0    -975563   0.000   1023.9
     3       0       0       0       0            0.0       0.0    648.6      3
   634.1   648.6   634.1            465.0    1437380   0.000   1011.0
     4       0       1      -1       1            0.0    115.0     -1.0      4
   461.3   466.6   461.3            215.0    -454183   0.000    212.2
     5       0       0       0       0            0.0       0.0    466.6      5
   461.3   466.6   461.3            215.0   -2735952   0.000   1278.2
     6       0       0       0       0            0.0       0.0    653.8      6
   644.5   653.8   644.5            200.0     581610   0.000    203.8
     7       0       0       0       0            0.0       0.0    547.6      7
   533.7   547.6   533.7            115.0   -1296919   0.000    959.2
     1       0       0      -1       1            0.0    215.0     -1.0      8
   240.0   240.0   248.6            215.0     925856   1.000    751.6
     4       0       0      -1       1            0.0    115.0     -1.0      9
   424.2   430.1   424.6            115.0    -442874    .164    212.2
     5       1       0       2       0            0.0       0.0    466.6     11
   461.3   466.6   464.4            215.0    2155206    .794   1278.2
```

C.. EQUIPMENT DATA

1	11	2	0	8	0	700	115.0	254.9	0.0	0.0	0.0	233843	29240	99393
2	1	8	202	9	0	-1377731	684.5	13.7	7654.1	0.0	1.0	18549	2204	7769
3	11	9	0	10	0	717	254.9	565.0	0.0	0.0	0.0	238590	29964	101541
4	1	10	202	11	0	-2071520	828.1	17.8	11508.4	0.0	1.0	21601	3314	9795
5	10	-1	0	-8	0	100	565.0	215.0	0.0	0.0	0.0	0	0	0
6	10	-4	0	-9	0	16	215.0	115.0	0.0	0.0	0.0	0	0	0
8	2	-3	201	0	0	4493258	146.9	34.7	249.6	0.0	1.0	6547	35946	37910
10	2	-6	201	0	0	1377391	47.0	32.8	76.5	0.0	1.0	3305	11019	12011
12	30	-8	0	0	0	1555873	-1.0	0.0	0.0	0.0	0.0	0	-94111	-94111
15	1	1	-2	12	0	-4432222	3239.9	6.6	0.0	0.0	1.0	64331	0	19299
16	1	3	203	0	0	-2394842	919.0	30.0	301.5	296.0	3.5	82186	264391	289046
17	1	4	203	0	0	-3926780	513.0	8.0	481.9	470.0	1.0	43146	78540	91484
20	11	5	0	13	0	292	215.0	531.2	0.0	0.0	0.0	114499	12194	46543
21	1	13	-5	0	-11	-4865113	3192.2	7.3	0.0	0.0	1.0	63573	0	19072
22	1	6	202	0	0	-1703878	861.5	5.3	9466.0	0.0	1.0	22295	2726	9415
27	11	7	0	15	0	344	115.0	222.4	0.0	0.0	0.0	130811	14365	53608
28	1	15	-7	0	0	-6628207	3598.3	5.5	0.0	0.0	1.0	69965	0	20989
29	1	11	203	16	0	-1744371	296.0	20.5	495.0	470.0	1.0	10350	34800	45150
31	2	-11	201	0	0	1317738	22.0	93.4	73.2	0.0	1.0	2097	10542	11171
34	1	12	202	0	0	-12725440	6958.8	5.1	70696.9	0.0	1.0	118584	20361	55936
39	1	16	-9	20	0	-1125983	224.0	25.0	0.0	0.0	1.0	8434	0	2530
47	1	20	203	24	0	-1847872	612.0	15.0	435.0	425.0	1.0	16969	64306	69397
53	1	24	203	0	0	-1515039	417.0	20.0	400.0	385.0	2.0	24946	115143	122627

D REFRIGERATION UNIT (RUNIT Section)

i) Input (to MAINR)

The data set begins with the usual title and system parameter cards. This is followed by previous refrigerant temperature level and cost information (X, Y vectors) which is required to compute the cost for streams purchased by the refrigeration unit for cold recovery.

The new refrigeration demand data (temperature levels and cooling loads, RLEV matrix) is preceded by a card specifying the number of levels for each refrigerant circuit (methane, ethylene and propane) and the total number of levels.

Finally the purchased stream information is read consisting of: i) the number of such streams and ii) stream properties and mole fraction vectors preceded by the circuit number within which the stream is first to be utilized for refrigerant cooling.

ii) Output (from RUNIT)

The output is divided into three sections, one for each refrigerant circuit (methane, ethylene and propane). Within each section there is a stream properties vector (SMRB) block followed by an equipment vector (EMR) block. Note that no stream control vectors are required and since each circuit uses a pure refrigerant no mole fraction vector is needed. Note also that the saturated liquid stream (stream 2 in both sections) is shown as being of unit flow for convenience in calculation. Its true flow is the total refrigerant circulation as shown for the final stream in both sections.

The final output consists of summarized information for each refrigerant level in the unit. This includes updated refrigerant unit cost values (in $/BTU) which serve as data points for creation of subsequent energy cost splines.

```
C*** HP RUNIT SECTION INPUT ***
 RUNIT - HP PLANT (8 SOLD)
 $PARLST
 AMORT=0.3,HRS=8000.,
 TWAT=535.,DTW=10.,CWAT=3.6E-5,TS=825.,HVS=1.00E3,CS=1.80E-2,CKWH=0.007,
 APPP=10.,APRR=10.,ARRR=5.,TRRR=310.,DTF(2)=0.,
 $END
C.. PREVIOUS REFRIGERATION LEVELS + COSTS -
    5
296.        0.0000133
345.        0.0000114
385.        0.0000096
425.        0.0000044
470.        0.0000024
C.. NEW REFRIGERATION DEMANDS -
    0     3     2     5
296.        2400000.
345.              0.
385.        1520000.
425.        1850000.
470.        5670000.
C.. PURCHASED STREAM INFORMATION -
    1
    2
240.0       240.0       248.6       215.0       925856.     1.000       751.6
.3393       .6495       .0104       .0008       .0000       .0000       .0000
```

C*** HP REFRIGERATION UNIT ***

C.. ETHYLENE SECTION STREAM VECTORS -

1	240.0	240.0	248.6	215.0	925856	1.000	751.6
2	435.0	435.0	435.0	263.9	-2839	0.000	1.0
3	435.0	435.0	371.6	263.9	-2882024	0.000	694.7
4	240.0	240.0	425.0	215.0	1835877	1.000	751.6
5	296.1	296.1	296.1	11.5	461059	1.000	419.2
6	435.0	435.0	347.4	263.9	-1938941	0.000	419.2
7	296.1	296.1	361.6	11.5	661070	1.000	419.2
8	385.2	385.2	371.6	106.9	385959	1.000	273.3
9	345.1	345.1	498.2	44.5	1148462	1.000	419.2
10	385.2	385.2	612.8	106.9	1643295	1.000	419.2
11	385.2	385.2	545.0	106.9	1319459	1.000	419.2
12	385.2	385.2	480.2	106.9	1705418	1.000	692.5
13	435.2	435.2	594.8	263.9	2446804	1.000	692.5
14	435.2	435.2	545.0	263.9	2032981	1.000	692.5

C.. ETHYLENE SECTION EQUIPMENT VECTORS -

1	1	2	1	3	4		-910021		
505.2		55.7	0.0		0	3.50	50911	0	15273
2	10	6	0	5	0		16		
263.9		11.5	0.0		0	0	0	0	0
3	1	3	5	6	7		-200011		
297.0		26.8	0.0		0	3.50	34962	0	10489
4	10	3	0	8	0		0		
263.9		106.9	0.0		0	0	0	0	0
5	11	7	0	9	0		213		
11.5		44.5	0.0		0	0	88638	8891	35483
6	11	9	0	10	0		216		
44.5		106.9	0.0		0	0	89731	9027	35946
7	1	10	202	11	0		-323837		
176.9		12.2	1799.1		0	1.00	7321	518	2714
8	21	11	8	12	0		100		
106.9		106.9	0.0		0	0	0	0	0
9	11	12	0	13	0		324		
106.9		263.9	0.0		0	0	124553	13525	50890
10	1	13	202	14	0		-413823		
275.3		9.6	2299.0		0	1.00	9545	662	3526
11	1	14	203	2	0		-4012117		
2666.2		11.2	0.0		425.0	1.00	55045	0	16514

C.. PROPANE SECTION STREAM VECTORS -

1	240.0	240.0	425.0	215.0	1835877	1.000	751.6
2	545.0	545.0	545.0	171.4	-960	0.000	1.0
3	545.0	545.0	535.3	174.4	-2765597	0.000	2207.8
4	240.0	240.0	535.0	215.0	2481729	1.000	751.6
5	425.2	425.2	425.2	18.2	4372435	1.000	1189.2
6	470.0	470.0	470.0	45.1	4385829	1.000	1025.2
7	470.0	470.0	497.3	45.1	5643455	1.000	1189.2
8	470.0	470.0	484.8	45.1	10029285	1.000	2214.4
9	555.2	555.2	598.2	174.4	13987886	1.000	2214.4

C.. PROPANE SECTION EQUIPMENT VECTORS -

1	1	2	1	3	4		-645852		
386.3		24.1	0.0		0	1.00	11697	0	3509
2	10	3	0	5	0		39		
174.4		18.3	0.0		0	0	0	0	0
3	10	3	0	6	0		25		
174.4		45.1	0.0		0	0	0	0	0
4	11	5	0	7	0		555		

```
  18.3      45.1      0.0         0         0  193344    23186    81189

 5   21        7    6        8    0            100
  45.1      45.1      0.0         0         0        0        0        0

 6   11        8    0        9    0           1729
  45.1     174.4      0.0         0         0  493529    72214   220272

C.. REFRIGERATION LEVEL DETAILS - TEMP,PRES,DEMAND,FLOW,UNIT COST -
   T,P,Q,F,C -
   296.0     11.5    2400000    419.2    0.0000138
   345.0     44.5          0      0.0    0.0000000
   385.0    106.9    1520000    273.4    0.0000095

   425.0     18.3    5862117   1189.2    0.0000043
   470.0     45.1    5670000   1025.2    0.0000025
```

III.2 Low Pressure Process Details

In this section data for the low pressure process case are presented. For this case only essential input data and output data for the optimal process configuration are presented. The figures correspond to case #3 in Table 8 (section 7.5). As the data format corresponds so closely to that for the previous high pressure case few notes of explanation are included with the data. Note that LPM and LPL refer to the medium and low temperature sub-processes as described in section 7.1.

*** LOW PRESSURE PROCESS CASE ***

```
C*** LP  COLSYS SECTION INPUT ***

C.. COLUMN SYSTEM DATA
 LP  COLUMN SYSTEM
 $PARLST
 AMORT=0.3,HRS=8000.,
 TWAT=535.,DTW=10.,CWAT=3.6E-5,TS=825.,HVS=1.00E3,CS=1.80E-2,CKWH=0.007,
 APPP=15.,APRR=10.,ARRR=5.,TRRR=310.,
 $END
C.. COLUMN NO.  1 DEETHANIZER
C..             2 DEMETHANIZER
C..             3 HP C2 SPLITTER
C..             4 LP C2 SPLITTER
C..             5 DEPROPANIZER
C..             6 C3 SPLITTER
 $KPMLST
 NIS=13,NKPM=6,
 KPM1=1,100,1,-2,-3,0,
 KPM2=2,100,2,-4,-5,0,
 KPM3=3,100,5,-6,-7,0,
 KPM4=4,100,7,-8,-9,0,
 KPM5=5,100,3,-10,-11,0,
 KPM6=6,100,10,-12,-13,0,
 $END
 $SMPLST
 SMPA1=1.,1.,1.,215.,0.,420.,2*0.,
 SMPA2=2.,0.,1.,0.,0.,270.,2*0.,
 SMPA9=9.,-1.,1.,0.,2.,0.,2*0.,
 SMPB1=2*0.,520.,115.,0.,1.,1500.,255.,495.,315.,210.,135.,45.,45.,0.,
 $END
 $EMILST
 EMI1=1.,100.,1.,1.2,0.162,0.0025,0.015,0.565,4.,5.,250.,24.,3*0.,
 EMI2=2.,100.,-1.,1.1,0.98,0.005,0.01,0.58,2.,3.,75.,24.,3*0.,
 EMI3=3.,100.,0.,1.1,0.96,0.030,0.450,0.540,3.,4.,65.,24.,3*0.,
 EMI4=4.,100.,-1.,1.1,0.96,0.04,0.01,0.94,3.,4.,20.,24.,3*0.,
 EMI5=5.,100.,0.,1.2,0.25,0.035,0.04,0.835,6.,7.,200.,24.,3*0.,
 EMI6=6.,100.,0.,1.2,0.90,0.08,0.075,0.76,5.,6.,115.,18.,3*0.,
 $END
```

C*** LP COLUMN EQUIPMENT VECTORS ***

1 100	0	1.00	396.5	0.73	11	9	3.8	0.60
1.00	14683	86854	0	26056				
2 100	0	-1.00	0.0	0.27	9	7	2.31	0.13
2.56	1495	49368	0	14809				
3 100	0	0.00	0.0	1.93	21	7	2.73	0.13
1.72	2830	63500	0	19050				
4 100	0	-1.00	0.0	2.49	17	22	2.33	0.09
1.72	3440	104832	0	31450				
5 100	0	0.00	0.0	0.48	7	10	1.91	0.27
1.00	2793	24773	0	7432				
6 100	0	0.00	0.0	6.14	29	56	3.93	0.31
1.00	21176	228661	0	68598				

```
C*** LPM  SMATCH SECTION INPUT ***

 LP PLANT - MED TEMP SECTION (PROPANE PSEUDO-SERVICE)
   1    1    1    1    1
 $PARLST
 AMORT=0.3,HRS=8000.,
 TWAT=535.,DTW=10.,CWAT=3.6E-5,TS=825.,HVS=1.00E3,CS=1.80E-2,CKWH=0.007,
 APPP=10.,APRR=10.,ARRR=5.,TRRR=310.,
 $END
    5    3    1    0    5    3
C.. HOT STREAM DATA
     1                        -2      1                        0.0    555.0
     2                         1                             250.0    420.0
     3                         0                               0.0    396.5
     4                         0                               0.0    553.9
     5                         0                               0.0    510.6
   555.2    555.2    610.0           174.4       6580  1.000      1.0
     0.00      0.00       0.00          0.00      0.00      1.00       0.00
   240.0    449.1    520.0           115.0    5185628  1.000   1500.0
   255.00    495.00    315.00        210.00    135.00     45.00      45.00
   240.0    423.9    423.9           250.0    4567941  1.000   2174.1
   257.90    583.16    684.32        627.13     18.74      2.83        .00
   553.9    560.4    560.4           200.0    1229635  1.000    265.0
     0.00      0.00       0.00          5.14    187.55     63.19       9.10
   510.6    513.6    513.6           115.0    3810960  1.000    985.1
     0.00      0.00       0.00         24.77    881.90     78.39       0.00

C.. COLD STREAM DATA
     1                         0                               0.0    603.6
     2                         0                               0.0    653.7
     3                         0                               0.0    547.4
   583.8    603.6    583.8           250.0    -346934  0.000   1101.7
      .00       .00       2.32         17.11    639.30    219.35     223.64
   644.3    653.7    644.3           200.0     702280  0.000    247.0
     0.00      0.00       0.00           .01     17.07     10.23     219.66
   533.6    547.4    533.6           115.0   -1297719  0.000    956.5
     0.00      0.00       0.00           .00     73.03    740.03     143.45

C.. REFRIGERATION LEVELS -
    5
305.
345.
385.
425.
470.
```

```
C*** LPM  OPTIMUM PLANT ***

C.. HOT STREAM DATA
  0.0000     0.0000      0.0000      0.0000      0.0000      1.0000      0.0000
   .1700      .3300       .2100       .1400       .0900       .0300       .0300
   .1186      .2682       .3148       .2885       .0086       .0013      0.0000
  0.0000     0.0000      0.0000       .0194       .7077       .2385       .0343
  0.0000     0.0000      0.0000       .0251       .8952       .0796      0.0000

   1        0      0     -2      1              0.0      0.0    555.0       1
 555.2    555.2   610.0                174.4     6580    1.000     1.0
   2        0      1      1      0              0.0    250.0    420.0       2
 240.0    449.1   520.0                115.0   5185628   1.000   1500.0
   3        0      0      0      0              0.0      0.0    396.5       3
 240.0    423.9   423.9                250.0   4567941   1.000   2174.1
   4        0      0      0      0              0.0      0.0    553.9       4
 553.9    560.4   560.4                200.0   1229635   1.000    265.0
   5        0      0      0      0              0.0      0.0    510.6       5
 510.6    513.6   513.6                115.0   3810960   1.000    985.1
   2        0      1      1      0              0.0    250.0    420.0       6
 240.0    484.9   620.6                250.0   6743676   1.000   1500.0
   2        0      0      1      0              0.0    250.0    420.0       7
 240.0    484.9   545.0                250.0   5417232   1.000   1500.0
   5        1      1      0      0              0.0      0.0    510.6       8
 555.5    557.9   575.1                219.2   4957913   1.000   1083.6
   2        1      0      1      0              0.0    250.0    420.0       9
 240.0    484.9   495.0                250.0   4578582   1.000   1500.0
   2        2      0      1      0              0.0    250.0    420.0      10
 240.0    484.9   445.0                250.0   2407796    .851   1500.0

C.. COLD STREAM DATA
  0.0000     0.0000       .0021       .0155       .5803       .1991       .2030
  0.0000     0.0000      0.0000       .0000       .0691       .0414       .8893
  0.0000     0.0000      0.0000      0.0000       .0764       .7737       .1500

   1        0      0      0      0              0.0      0.0    603.6       1
 583.8    603.6   583.8                250.0   -346934   0.000   1101.7
   2        0      0      0      0              0.0      0.0    653.7       2
 644.3    653.7   644.3                200.0    702280   0.000    247.0
   3        0      0      0      0              0.0      0.0    547.4       3
 533.6    547.4   533.6                115.0  -1297719   0.000    956.5

C.. EQUIPMENT DATA
   1   11        2      0       6      0             680
 115.0      250.0       0.0          0.0      0.0  228460       28422       96960

   2    1        6    202       7      0        -1326444
 674.8       13.4    7369.1          0.0      1.0   18337        2122        7623

   3    2       -1    201       0      0         6762353
 175.5       47.2     375.7          0.0      1.0    7285       54099       56284

   4    2       -2    201       0      0         1669474
  56.9       32.9      92.7          0.0      1.0    3707       13356       14468

   7    1        3    203       0      0        -4937518
1756.0       16.0     396.5        385.0      2.0   39418      387120      398945

   8    1        4    202       0      0        -1594803
 746.5        5.7    8860.0          0.0      1.0   19881        2552        8516

  11   11        5      0       8      0             334
 115.0      219.2       0.0          0.0      0.0  127886       13971       52337

  12    1        8     -3       0      0        -6608847
3484.6        5.7       0.0          0.0      1.0   68191           0       20457

  13    1        7    203       9      0         -838650
 149.0       21.0     495.0        470.0      1.0   10200       22140       25200

  15    1        9    203      10      0        -2170786
 490.0       21.0     445.0        425.0      1.0   14187       90305       94561

  16    1       10    203       0      0        -1382487
 238.0       28.0     420.0        385.0      2.0    8757      108427      111054
```

```
C*** LPL  SMATCH SECTION INPUT ***

 LP PLANT - LOW TEMP SECTION - NEW 2/1 MATCH SET
    0    1    0    1    1
 $PARLST
 AMORT=0.3,HRS=8000.,
 TWAT=535.,DTW=10.,CWAT=3.6E-5,TS=825.,HVS=1.00E3,CS=1.80E-2,CKWH=0.007,
 APPP=10.,APRR=10.,ARRR=7.,TRRR=400.,
 $END
    4    5    1    0    4    5
C.. HOT STREAM DATA
     1                  -2      1                 0.0   555.0
     2                   1                      250.0   270.0
     3                   0                        0.0   350.1
     4                   0                        0.0   315.1
   555.2   555.2   610.0          174.4      6580  1.000     1.0
     0.00      0.00     0.00        0.00      0.00     1.00       0.00
   240.0   396.5   396.5          250.0   2493182  1.000  1274.7
   255.00    495.00   314.62      206.48      3.17      .39        .00
   350.1   362.1   362.1           65.0    783953  1.000   533.6
     0.00     16.02   501.93       15.67       .00     0.00       0.00
   315.1   316.1   316.1           20.0    211030  1.000   171.6
     0.00      0.00   165.10        6.50       .00     0.00       0.00

C.. COLD STREAM DATA
     1                   0                   75.0   384.3
     2                   0                    0.0   386.0
     3                   0                    0.0   345.8
     4                  -1      1             0.0    -1.0
     5                  -1      1             0.0    -1.0
   373.3   384.3   373.3           75.0  -1657187  0.000   411.0
     .00      4.30    243.13      161.28      2.26     0.00       0.00
   379.1   386.0   379.1           65.0  -2059954  0.000   523.6
     0.00       .00   212.02      307.12      4.42     0.00       0.00
   342.7   345.8   342.7           20.0  -2054578  0.000   439.1
     0.00      0.00     4.80      428.20      6.30     0.00       0.00
   342.7   345.8   342.7           20.0   -933912  0.000   199.6
     0.00      0.00     2.18      194.55      2.88     0.00       0.00
   240.0   240.0   250.0           75.0    953000  1.000   749.7
   255.00    489.50     5.20        0.00      0.00     0.00       0.00

C.. REFRIGERATION LEVELS -
    7
245.
275.
305.
345.
385.
425.
470.
```

```
C*** LPL  OPTIMUM PLANT ***

C.. HOT STREAM DATA
   0.0000     0.0000     0.0000     0.0000     0.0000     1.0000     0.0000
    .2000      .3883      .2468      .1620      .0025      .0003     0.0000
   0.0000      .0300      .9406      .0294     0.0000     0.0000     0.0000
   0.0000     0.0000      .9621      .0379     0.0000     0.0000     0.0000

     1       0      0       -2      1             0.0     0.0   555.0       1
   555.2   555.2   627.0              174.4       6580  1.000      1.0
     2       0      0        1      0             0.0   250.0   270.0       2
   240.0   396.5   396.5              250.0    2493182  1.000   1274.7
     3       0      0        0      0             0.0     0.0   350.1       3
   350.1   362.1   362.1               65.0     783953  1.000    533.6
     4       0      0        0      0             0.0     0.0   315.1       4
   315.1   316.1   316.1               20.0     211030  1.000    171.6
     2       1      0        1      0             0.0   250.0   270.0       5
   240.0   396.5   380.3              250.0    1185677   .817   1274.7
     3       1      1        0      0             0.0     0.0   350.1       6
   385.9   395.9   426.6              132.0    1119526  1.000    587.0
     3       2      0        0      0             0.0     0.0   385.9       7
   385.9   395.9   388.9              132.0   -1959108   .070    587.0
     3       1      0        0      0             0.0     0.0   350.1       8
   350.1   362.1   345.9               65.0   -2110002   .076    533.6
     4       1      1        0      0             0.0     0.0   315.1      11
   351.4   355.9   402.0               55.2     346612  1.000    188.8
     2       2      0        1      0             0.0   250.0   270.0      15
   240.0   396.5   350.1              250.0    -503950   .603   1274.7
     2       3      0        1      0             0.0   250.0   270.0      19
   240.0   396.5   315.0              250.0   -1899526   .439   1274.7
C.. COLD STREAM DATA
   0.0000      .0105      .5916      .3924      .0055     0.0000     0.0000
   0.0000     0.0000      .4049      .5866      .0084     0.0000     0.0000
   0.0000     0.0000      .0109      .9752      .0143     0.0000     0.0000
   0.0000     0.0000      .0109      .9747      .0144     0.0000     0.0000
    .3401      .6529      .0069     0.0000     0.0000     0.0000     0.0000

     1       0      0        0      0             0.0    75.0   384.3       1
   373.3   384.3   373.3               75.0   -1657187  0.000    411.0
     2       0      0        0      0             0.0     0.0   386.0       2
   379.1   386.0   379.1               65.0   -2059954  0.000    523.6
     3       0      0        0      0             0.0     0.0   345.8       3
   342.7   345.8   342.7               20.0   -2054578  0.000    439.1
     4       0      0       -1      1             0.0     0.0    -1.0       4
   342.7   345.8   342.7               20.0    -933912  0.000    199.6
     5       0      0       -1      1             0.0     0.0    -1.0       5
   240.0   240.0   250.0               75.0     953000  1.000    749.7
     1       1      0        0      0             0.0    75.0   384.3       6
   373.3   384.3   380.2               75.0    -338831   .547    411.0
     3       1      0        0      0             0.0     0.0   345.8       8
   342.7   345.8   343.1               20.0    -880927   .407    439.1
     3       2      0        0      0             0.0     0.0   345.8      10
   342.7   345.8   345.8               20.0     846707  1.000    439.1
C.. EQUIPMENT DATA
   1   30       -4      0         0     0          1738220
   343.0     535.0        0.0         0.0       0.0              0   -147345   -147345

   2   30       -5      0         0     0          1547976
   250.0     535.0        0.0         0.0       0.0              0   -135146   -135146

   5    1        2     -1         5    -6         -1307505
   930.1       8.1        0.0         0.0       2.0      47410                0     14223

   7   11        3      0         6     0              112
    65.0     132.0        0.0         0.0       0.0      52944             4692     20575

   8    1        6     -2         7     0         -3092540
  2725.8       7.0        0.0         0.0       2.0     112052                0     33616

  14   11        4      0        11     0               50
    20.0      55.2        0.0         0.0       0.0      27709             2088     10401

  15    1       11     -3         0    -8         -1204248
```

915.2		10.3	0.0	0.0	2.0	46800	0	14040
16	1	301	-6	0	0	-1091359		
58.6		83.3	150.8	555.0	2.0	7546	0	2264
22	1	5	-8	15	-10	-1689628		
811.1		16.0	0.0	0.0	2.0	42491	0	12747
23	1	7	203	0	0	-234684		
45.0		31.0	385.9	345.0	2.0	6472	22905	24847
30	1	15	203	19	0	-1395937		
479.0		27.0	315.0	305.0	3.5	48807	149649	164291
34	1	19	203	0	0	-1507047		
276.0		66.0	270.0	245.0	3.5	33444	295381	305414

```
C*** LP  REFRIGERATION UNIT ***

C.. METHANE SECTION STREAM VECTORS -
   1   240.0    250.0    250.0          75.0    953000   1.000    749.7
   2   315.0    315.0    315.0         400.0     -1070   0.000      1.0
   3   315.0    315.0    292.0         400.0  -1008000   0.000    678.0
   4   240.0    240.0    305.0          75.0   1236000   1.000    749.7
   5   315.0    315.0    275.0         400.0  -1195000   0.000    678.0
   6   245.0    245.0    245.0          75.0    648000   1.000    678.0
   7   245.0    245.0    245.0          75.0    101400   1.000    106.0
   8   245.0    245.0    245.0          75.0    547000   1.000    572.0
   9   245.0    245.0    282.0          75.0    734000   1.000    572.0
  10   245.0    245.0    245.0          75.0    101400   1.000    106.0
  11   245.0    245.0    275.0          75.0    835000   1.000    678.0
  12   315.0    315.0    408.0         400.0   1415000   1.000    678.0

C.. METHANE SECTION EQUIPMENT VECTORS -
   1    1         2      1         3       4           -283000
   210.0       27.0       0.0              0       3.50     28800          0       8630

   2    1         3      8         5       9           -187000
   178.0       28.0       0.0              0       3.50     25900          0       7770

   3   10         5      0         6       0                11
   400.0       75.0       0.0              0          0         0          0          0

   4   20         6      0         7       8                16
   84.0         0.0       0.0              0          0         0          0          0

   5   21         9     10        11       0               100
   75.0        75.0       0.0              0          0         0          0          0

   6   11        11      0        12       0               254
   75.0       400.0       0.0              0          0    103000      10600      41500

   7    1        12    203         2       0          -2140000
 1352.0        25.0       0.0          305.0       2.00     63400     230000     249000


C.. ETHYLENE SECTION STREAM VECTORS -
   1   240.0    240.0    305.0          75.0    236000   1.000    749.7
   2   342.7    345.8    342.7          20.0   -933912   0.000    199.6
   3   435.0    435.0    435.0         263.9     -2839   0.000      1.0
   4   435.0    435.0    390.3         263.9  -6552384   0.000    173.5
   5   240.0    240.0    425.0          75.0   1870523   1.000    749.7
   6   435.0    435.0    352.7         263.9  -7828432   0.000   1732.5
   7   342.7    345.8    345.3          20.0    345140   0.970    199.6
   8   305.0    305.0    305.0          15.2    682762   1.000    594.1
   9   435.0    435.0    339.0         263.9  -2847238   0.000    594.1
  10   305.0    305.0    342.7          15.2    845345   1.000    594.1
  11   345.1    345.1    345.1          44.5     54409   1.000     40.0
  12   385.2    385.2    352.7         106.9   1356813   1.000   1098.4
  13   345.1    345.1    442.0          44.5   1305327   1.000    594.1
  14   345.1    345.1    436.1          44.5   1359736   1.000    634.1
  15   385.2    385.2    536.5         106.9   1936957   1.000    634.1
  16   385.2    385.2    423.4         106.9   3293770   1.000   1732.5
  17   435.2    435.2    524.5         263.9   4667613   1.000   1732.5

C.. ETHYLENE SECTION EQUIPMENT VECTORS -
   1    1         3      1         4       5          -1634523
 1162.8        34.0       0.0              0       3.50     99194          0      29758

   2    1         4      2         6       7          -1276048
   587.7       21.1       0.0              0       2.00     32834          0       9850

   3   10         9      0         8       0                11
   263.9       15.2       0.0              0          0         0          0          0

   4    1         6      8         9      10           -162584
   310.9       20.0       0.0              0       3.50     35933          0      10780

   5   10         6      0        11       0                 3
   263.9       44.5       0.0              0          0         0          0          0

   6   10         6      0        12       0                 0
```

```
 263.9     106.9        0.0           0          0          0          0          0

 7   11       10    0           13    0                201
  15.2      44.5        0.0           0          0     84585       8391      33766

 8   21       13   11           14    0                100
 44.5       44.5        0.0           0          0          0          0          0

 9   11       14    0           15    0                252
 44.5      106.9        0.0           0          0    101650      10530      41025

10   21       15   12           16    0                100
 106.9     106.9        0.0           0       1.00          0          0          0

11   11       16    0           17    0                600
 106.9     263.9        0.0           0          0    206062      25062      86881

12    1       17  203            3    0           -9618815
6922.1       9.5                  425.1       1.00    118084          0      35425

C.. PROPANE SECTION STREAM VECTORS -
    1   240.0   240.0   425.0             75.0   1870523   1.000   749.7
    2   342.7   345.8   345.2             20.0    345140   0.970   199.6
    3   555.0   555.0   555.0            174.4      -656   0.000     1.0
    4   555.0   555.0   546.5            174.4  -2410690   0.000  2624.1
    5   240.0   240.0   545.0             75.0   2560716   1.000   749.7
    6   555.0   555.0   540.7            174.4  -2873634   0.000  2624.1
    7   342.7   345.8   536.5             20.0    808084   1.000   199.6
    8   425.2   425.2   425.2             18.3   9083356   1.000  2470.5
    9   470.0   470.0   470.0             45.1    668804   1.000   156.3
   10   470.0   470.0   497.3             45.1  11723790   1.000  2470.5
   11   470.0   470.0   495.7             45.1  12392594   1.000  2626.9
   12   555.2   555.2   626.9            174.4  16976226   1.000  2476.1

C.. PROPANE FLOW TO PSEUDO-SERVICE -  150.8

C.. PROPANE SECTION EQUIPMENT VECTORS -
  1   1        3    1            4    5            -690193
 386.4      25.7        0.0           0       1.00     11698          0       3510

  2   1        4    2            6    7            -462944
 197.7      51.4        0.0           0       2.00     15651          0       4695

  3  10        6    0            8    0                 41
 174.4      18.3        0.0           0          0          0          0          0

  4  10        6    0            9    0                 27
 174.4      45.1        0.0           0          0          0          0          0

  5  11        8    0           10    0               1153
  18.3      45.1        0.0           0          0    352786      48167     154003

  6  21       10    9           11    0                100
  45.1      45.1        0.0           0          0          0          0          0

  7  11       11    0           12    0               2454
  45.1     174.4        0.0           0          0    661031     102533     300842

  8   1       12  202            3    0          -18661564
9162.9       7.2 103675.4             0       1.00    147784      29859      74194
```

REFERENCES FOR THE APPENDICES

1. Lee, K. F., Masso, A. H. and Rudd, D. F.,
"Branch and Bound Synthesis of Integrated Process Designs", I & EC Fund., 9 No. 1, 48 (1970).

2. Motard, R. L., Lee, H. M., Barkely, R. W. and Ingels, D. M.,
"CHESS, Chemical Engineering Simulation System, System Guide", Tech. Publ. Co., 4375 Harvest Lane, Houston, Texas (1968).

3. Hengstebeck, R. J.,
"An Improved Shortcut for Calculating Difficult Multicomponent Distillations", Chem. Eng., 76 No. 1, 115 (1969).

4. Stoppel, A. E.,
I & EC, 38,1271 (1946).

5. Bolles, W. L.,
"Optimum Bubble-Cap Tray Design, Part IV", Petr. Proc., 109 (May, 1956).

6. Peters, M. S. and Timmerhaus, K. D.,
"Plant Design and Economics for Chemical Engineers, 2nd Edn.", McGraw-Hill (1968).

7. Edmister, W. C.,
"Applied Hydrocarbon Thermodynamics", Gulf Publ. Co. (1961).