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
SYNTHESIS OF OPTIMAL ENERGY EXCHANGE NETWORKS USING DISCRETE METHODS

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

## PART I  THEORY AND PROGRAM SYSTEM

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>General</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Background</td>
<td>2</td>
</tr>
<tr>
<td>1.2.1</td>
<td>Simulation</td>
<td>2</td>
</tr>
<tr>
<td>1.2.2</td>
<td>Synthesis</td>
<td>3</td>
</tr>
<tr>
<td>1.2.3</td>
<td>Optimization</td>
<td>6</td>
</tr>
<tr>
<td>1.3</td>
<td>Synthesis - Study Philosophy and Objectives</td>
<td>8</td>
</tr>
<tr>
<td>1.3.1</td>
<td>Study Philosophy</td>
<td>8</td>
</tr>
<tr>
<td>1.3.2</td>
<td>Study Objectives</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>THEORY</td>
<td>12</td>
</tr>
<tr>
<td>2.1</td>
<td>Branch and Bound</td>
<td>12</td>
</tr>
<tr>
<td>2.2</td>
<td>Process Decomposition</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>DESIGN CONSIDERATIONS</td>
<td>24</td>
</tr>
<tr>
<td>3.1</td>
<td>Unit Operations</td>
<td>24</td>
</tr>
<tr>
<td>3.2</td>
<td>Heuristic Development</td>
<td>25</td>
</tr>
<tr>
<td>3.3</td>
<td>Stream Energy Pricing for Process Decomposition</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>PROGRAM SYSTEM</td>
<td>32</td>
</tr>
<tr>
<td>4.1</td>
<td>General</td>
<td>32</td>
</tr>
<tr>
<td>4.2</td>
<td>Program Functions</td>
<td>34</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Task Identification (COLSYS)</td>
<td>34</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Stream Processing Path Generation (SMATCH)</td>
<td>35</td>
</tr>
</tbody>
</table>

(iii)
4.2.3 Stream Energy Costing (ENERGY)
4.2.4 Selection of Optimal Network Configuration (BRBND)
4.2.5 Physical Properties Calculation
4.2.6 Equipment Routines
4.3 System Data Structures
4.4 Programming and Operating Details

PART II APPLICATION STUDIES

Chapter 5 ETHYLENE PLANT - PROCESS DESCRIPTION AND CONSIDERATIONS
5.1 Ethylene Plant Description
  5.1.1 Cracking
  5.1.2 Feed Purification
  5.1.3 Product Recovery
5.2 Process Energy Considerations
5.3 Major Process Assumptions
5.4 Refrigeration Unit (RUNIT)

Chapter 6 HIGH PRESSURE PROCESS
6.1 Process Considerations and Problem Computation
6.2 Entropy Aspects
6.3 Branching Problem Selection
6.4 Stream Pricing and Energy Considerations

Chapter 7 LOW PRESSURE PROCESS
7.1 Process Considerations and Problem Strategy
7.2 Pseudo-Service Stream Usage
7.3 Problem Computation

(iv)
7.4 Modifications to Optimal Process Configuration

7.5 Comparison Between High and Low Pressure Processes

Chapter 8 PHYSICAL PROPERTIES CALCULATION

Chapter 9 CONCLUSIONS AND RECOMMENDATIONS

9.1 Conclusions

9.2 Recommendations

9.2.1 Improvements to Present System

9.2.2 Extension of Applications of Present System

9.2.3 Wider Extensions

9.3 Contributions

NOMENCLATURE

REFERENCES
# APPENDICES

## APPENDIX I DERIVATIONS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>I.1</td>
<td>Stream Energy Value Integration</td>
<td>112</td>
</tr>
<tr>
<td>I.2</td>
<td>Approximate Expression for Heat Exchanger Entropy Increase</td>
<td>114</td>
</tr>
</tbody>
</table>

## APPENDIX II PROGRAM SYSTEM

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>II.1</td>
<td>Program Descriptions and Listings</td>
<td>116</td>
</tr>
<tr>
<td>II.1.1</td>
<td>Task Identification</td>
<td>116</td>
</tr>
<tr>
<td>II.1.2</td>
<td>Stream Processing Path Generation</td>
<td>118</td>
</tr>
<tr>
<td>II.1.3</td>
<td>Selection of Optimal Network Configuration</td>
<td>127</td>
</tr>
<tr>
<td>II.1.4</td>
<td>Refrigeration Unit</td>
<td>134</td>
</tr>
<tr>
<td>II.1.5</td>
<td>Physical Properties</td>
<td>144</td>
</tr>
<tr>
<td>II.1.6</td>
<td>Equipment Routines</td>
<td>164</td>
</tr>
<tr>
<td>II.2</td>
<td>System Data Structures</td>
<td>176</td>
</tr>
</tbody>
</table>

## APPENDIX III CASE STUDY AND PROCESS DETAILS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>III.1</td>
<td>High Pressure Case Study and Process Details</td>
<td>223</td>
</tr>
<tr>
<td>III.2</td>
<td>Low Pressure Process Details</td>
<td>247</td>
</tr>
</tbody>
</table>

## REFERENCES FOR THE APPENDICES

(vi)
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Branch and Bound - Branching Strategy</td>
</tr>
<tr>
<td>2</td>
<td>Example First Level Branching Strategy</td>
</tr>
<tr>
<td>3</td>
<td>Sub-process Interaction</td>
</tr>
<tr>
<td>4</td>
<td>Unit Operations and Sub-process Procedures</td>
</tr>
<tr>
<td>5</td>
<td>Program System Structure</td>
</tr>
<tr>
<td>6</td>
<td>Ethylene Plant Typical Schematic</td>
</tr>
<tr>
<td>7</td>
<td>Product Separation Schemes</td>
</tr>
<tr>
<td>8</td>
<td>Typical Cascade Refrigeration Unit</td>
</tr>
<tr>
<td>9</td>
<td>High Pressure Ethylene Plant - Separation Scheme</td>
</tr>
<tr>
<td>10</td>
<td>Problem Solution Sequence</td>
</tr>
<tr>
<td>11</td>
<td>Optimal Process Configuration (H.P.)</td>
</tr>
<tr>
<td>12</td>
<td>Refrigeration Unit for High Pressure Process</td>
</tr>
<tr>
<td>13</td>
<td>Effect of &quot;Discount&quot; Parameter on Optimal Process Cost</td>
</tr>
<tr>
<td>14</td>
<td>Energy Cost Relations</td>
</tr>
<tr>
<td>15</td>
<td>Low Pressure Plant - Separation Scheme</td>
</tr>
<tr>
<td>16</td>
<td>Configuration of Optimal Low Pressure Process Network</td>
</tr>
<tr>
<td>17</td>
<td>Refrigeration Unit for Low Pressure Process</td>
</tr>
<tr>
<td>18</td>
<td>Process Configuration Described by Baldus and Linde</td>
</tr>
<tr>
<td>19</td>
<td>Modified Low Temperature Section</td>
</tr>
</tbody>
</table>

(vii)
<table>
<thead>
<tr>
<th>TABLE</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>General System Parameters</td>
<td>39</td>
</tr>
<tr>
<td>2</td>
<td>Distillation Column Parameters</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>Distribution of Losses in an Air Separation Process</td>
<td>51</td>
</tr>
<tr>
<td>4</td>
<td>Process Feed Details</td>
<td>60</td>
</tr>
<tr>
<td>5</td>
<td>High Pressure Process Operating Conditions</td>
<td>61</td>
</tr>
<tr>
<td>6</td>
<td>Effect of Entropy Increase Parameter (DENMX) on Problem Size</td>
<td>68</td>
</tr>
<tr>
<td>7</td>
<td>Low Pressure Process Operating Conditions</td>
<td>80</td>
</tr>
<tr>
<td>8</td>
<td>Comparison Between High and Low Pressure Processes</td>
<td>91</td>
</tr>
</tbody>
</table>
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 DDesign 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 tempera-
ture 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 interaction. 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 Sirola 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 Sirola 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 inter-
action and complexity of equipment interconnections. Energy exchange net-
works 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).
\[ O_B(D_A) \geq O_A(D_A) \]  
\( (1) \)

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).

\[ O_B(D_B) = O_A(D_B) \]  
\( (2) \)

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
\[ N = \prod_{i=1}^{m} n_i \]  

(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 \((10 \times 5 \times 7 \times 12)\) 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 \((1 \times 2 \times 4 \times 3)\) 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, \(m \in M\). 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, $X$, 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, $X$, 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

$$\text{Minimize } F(M)$$

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

$$\text{Minimize } F = [f_1(m_1) + f_2(m_2)]_{m_1, m_2, \in M}$$

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

Consider the transfers between sub-processes

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

and

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

$$\text{Minimize } f'_1 = f_1(m_1) + \sum_{i=1}^{n_1} P_{1i} X_{1i} - \sum_{j=1}^{n_2} P_{2j} X_{2j} \quad (7a)$$

and

$$\text{Minimize } f'_2 = f_2(m_2) + \sum_{j=1}^{n_2} P_{2j} X_{2j} - \sum_{i=1}^{n_1} P_{1i} X_{1i} \quad (7b)$$

noting that

$$f'_1 + f'_2 = F \quad (8)$$

provided that the flows, $X$, are continuous across the sub-process boundaries.

For a given set of prices, $P$, 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 product of those for the sub-processes. With the correct set of prices, $\hat{P}$, 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

$$\hat{f'_1} + \hat{f'_2} = F^* \quad (9)$$

where

$$\hat{f'_1} = \min_{m_1} f'_1(\hat{P}), \hat{f'_2} = \min_{m_2} f'_2(\hat{P}) \quad (10)$$
and $F^*$ is the overall process optimum, the solution to the original problem, (4).

The problem then is to adjust $P$ in such a way as to move towards $\hat{P}$. As Lasdon\(^{(13)}\) has shown, for $X$ continuous, $P$ can be adjusted by deliberately creating a discontinuity in $X$ between sub-processes and introducing $X$ as additional decision variables for the sub-processes. Then $P$ is adjusted to reduce excess supplies or demands for $X$. 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 $X$. 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 $X$ 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 $X$ 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, $P$, 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, $P$; $F$ in fact is piecewise constant in $P$. 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, \( \Delta X \). Thus there will in general be a certain range of \( P \) around \( \hat{P} \) within which the overall optimal solution \( P^* \) 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 \( P \) 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 \( P \) 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**

1) 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**

1) Set a minimum temperature of approach for heat exchange.
   
   This is a practical limitation imposed by process equipment.

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

3) Limit the pressure ratio for a single compression stage.
   
   This again is a practical equipment limitation.

**Stream Matching**

1) 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 especially 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.,

\[
\text{Price/NTU} = \text{pr}(T)
\]

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

\[ \epsilon = \Delta H - T_0 \Delta S \]  

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

\[ \frac{\epsilon}{B/\Delta U} = \frac{T_0 - T}{T}, \text{ the Carnot fraction} \]

In the present case the sink temperature, \( T_0 \), 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, \( \delta \), 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

\[ p(T-\delta). \]  

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 $\delta$ itself should be positive to reflect irreversibility, it is in fact a parameter representing relative 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

$$
P = \int_{T_1}^{T_2} p r(\theta + \delta) \left( \frac{dH}{d\theta} \right) d\theta
$$

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

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

The $\delta$ 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 $\delta$ 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, $P$, may become serious. In this case it is suggested that a single $\delta$ 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 $\delta$s. This introduces the possibility of
missing some solutions. However as the optimum solution has been shown to be constant over a certain range in \( P \) and the expected variation in \( s \) 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 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 as modified by Grayson and Streed. 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.

1) 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

<table>
<thead>
<tr>
<th>Equipment Costs</th>
<th>[Installed Capital Cost = ( a \times (size^b \times \text{Lang factor} \times f) )]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equipment</td>
<td>a</td>
</tr>
<tr>
<td>Heat Exchanger</td>
<td>82</td>
</tr>
<tr>
<td>Heat Exchanger</td>
<td>25</td>
</tr>
<tr>
<td>Compressor</td>
<td>480</td>
</tr>
<tr>
<td>Compressor Motor</td>
<td>34</td>
</tr>
</tbody>
</table>

Material Cost Factors for Heat Exchangers

- Down to \(-50^\circ F\): Carbon steel, 1.0
- \(-50^\circ F\) - \(-150^\circ F\): Nickel steel, 2.0
- Below \(-150^\circ F\): Stainless steel, 3.5
- Amortization fraction: 0.3/year

Service Costs

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

Other Parameters

- Minimum Exchanger Approach: \(10^\circ \text{F}\)
- Maximum Compressor Pressure Ratio per Stage: 4.0
Table 2

Distillation Column Parameters

<table>
<thead>
<tr>
<th>Equipment Costs</th>
<th>[Installed Capital Cost = (a \times (size)^b \times \text{Lang factor} \times f)]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>a</td>
<td>Size</td>
</tr>
<tr>
<td>Column Shell</td>
<td>14.5</td>
</tr>
<tr>
<td>Trays</td>
<td>48.0</td>
</tr>
</tbody>
</table>

Material Cost Factors and Stresses

<table>
<thead>
<tr>
<th>Condition</th>
<th>Material</th>
<th>Cost Factor</th>
<th>Stress (psi)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Down to -50°F</td>
<td>Carbon Steel</td>
<td>1.0</td>
<td>13750</td>
</tr>
<tr>
<td>-50°F to -150°F</td>
<td>Nickel Steel</td>
<td>2.0</td>
<td>16000</td>
</tr>
<tr>
<td>Below -150°F</td>
<td>Stainless Steel</td>
<td>3.5</td>
<td>18750</td>
</tr>
</tbody>
</table>

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. 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 C₃ and C₄ 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 C₂ splitter and 100 psia in the C₃ splitter. A typical plant is described by Aalund(28).
A) DEMETHANIZER FIRST

B) DEETHANIZER FIRST

C) DEPROPAIZER FIRST

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

<table>
<thead>
<tr>
<th>Source</th>
<th>Power Consumption (%)</th>
<th>Loss (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compressor Irreversibility</td>
<td>42</td>
<td>-</td>
</tr>
<tr>
<td>Column Irreversibility</td>
<td>20</td>
<td>52</td>
</tr>
<tr>
<td>Heat Exchange Irreversibility</td>
<td>9</td>
<td>23</td>
</tr>
<tr>
<td>Heat Inleak</td>
<td>7</td>
<td>15</td>
</tr>
<tr>
<td>Expansion Valves</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Turbine Irreversibility</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Reversible Separation Work</td>
<td>18</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>
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 \( \frac{R}{R_{MIN}} \). 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 C₂ 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 %) -

<table>
<thead>
<tr>
<th>Compound</th>
<th>Mole %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hydrogen</td>
<td>17</td>
</tr>
<tr>
<td>Methane ($C_1$)</td>
<td>33</td>
</tr>
<tr>
<td>Ethylene ($C_2$)</td>
<td>21</td>
</tr>
<tr>
<td>Ethane ($C_2O$)</td>
<td>14</td>
</tr>
<tr>
<td>Propylene ($C_3$)</td>
<td>9</td>
</tr>
<tr>
<td>Propane ($C_3O$)</td>
<td>3</td>
</tr>
<tr>
<td>Butane ($C_4$)</td>
<td>3</td>
</tr>
</tbody>
</table>

Total Flow: 1500 lb moles/hr.
Pressure: 115 psia
Temperature: 60°F
<table>
<thead>
<tr>
<th>Column Conditions</th>
<th>Column</th>
<th>Pressure (psia)</th>
<th>R/R_{\text{Min}}</th>
<th>Key Splits (Mole fractions)</th>
<th>Keys</th>
<th>Overhead</th>
<th>Bottom</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Demethanizer</td>
<td>565</td>
<td>1.1</td>
<td>C_1</td>
<td>0.65</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C_2^-</td>
<td>0.01</td>
<td>0.43</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deethanizer</td>
<td>465</td>
<td>1.2</td>
<td>C_2^o</td>
<td>0.39</td>
<td>0.015</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C_3^-</td>
<td>0.025</td>
<td>0.47</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C_2 Splitter*</td>
<td>215</td>
<td>1.2</td>
<td>C_2^-</td>
<td>0.96</td>
<td>0.01</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C_2^o</td>
<td>0.02</td>
<td>0.93</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Depropanizer</td>
<td>200</td>
<td>1.2</td>
<td>C_3^o</td>
<td>0.25</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C_4</td>
<td>0.04</td>
<td>0.84</td>
<td></td>
</tr>
<tr>
<td></td>
<td>C_3 Splitter</td>
<td>115</td>
<td>1.2</td>
<td>C_3^-</td>
<td>0.90</td>
<td>0.08</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>C_3^o</td>
<td>0.08</td>
<td>0.76</td>
<td></td>
</tr>
</tbody>
</table>

**Additional Stream Specifications**

- Demethanizer feed temperature: -60°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-BRBN 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
**KEY**

- S  STEAM
- W  WATER
- P  PROPANE REFRIGERANT
- E  ETHYLENE REFRIGERANT
- SR  SALES TO REFRIGERATION UNIT
- PR  PROPANE VAPOR FROM REFR. UNIT
- ACTIVE STREAMS NUMBERED 1-7  HOT
  -1-7  COLD

**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 C₂ and C₃ 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 \( \text{DENMX} \) 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 \( \text{DENMX} \) 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

\[
\text{DENMX} = \frac{a}{T^2}
\] (15)
<table>
<thead>
<tr>
<th>DENMX ( (X_{10^5}/R) )</th>
<th>Total No. of Streams</th>
<th>No. of Equipments</th>
<th>No. of Processing Paths</th>
<th>SMATCH Time (seconds)</th>
<th>No. of Path Combinations</th>
<th>Max. Size Sorting Problem</th>
<th>BRBND Time (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>40</td>
<td>53</td>
<td>37</td>
<td>10</td>
<td>0.18 * 10^6</td>
<td>34</td>
<td>7</td>
</tr>
<tr>
<td>30</td>
<td>42</td>
<td>57</td>
<td>40</td>
<td>12</td>
<td>0.58 * 10^6</td>
<td>22</td>
<td>10</td>
</tr>
<tr>
<td>40</td>
<td>47</td>
<td>66</td>
<td>48</td>
<td>18</td>
<td>3.20 * 10^6</td>
<td>78</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 6

Effect of Maximum Entropy Change Parameter (DENMX) on Problem Size
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 \( \text{DENX} \), 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, \( \text{DENX} \), 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 \([P = \text{Primary}, S = \text{Secondary}, T = \text{Tertiary}]\).

<table>
<thead>
<tr>
<th>Branching Level</th>
<th>Problem Type (Stream Match)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(P/P) matches</td>
</tr>
<tr>
<td>2</td>
<td>(P/S) and (S/S) matches</td>
</tr>
<tr>
<td>3</td>
<td>(P/T), (S/T) and (T/T) matches</td>
</tr>
</tbody>
</table>

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 $\delta$ can be expected to reflect this single extra degradation step and thus be of the order of the minimum exchanger approach temperature ($\Delta T_{\text{Min}}$). This is shown to be the case in Figure 13 where the optimal process cost is shown as a function of $\delta$ (expressed as a multiple of $\Delta T_{\text{Min}}$). It can be seen that any positive value of $\delta$ 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\(^\text{\ref{18}}\) 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 \(15^\circ\text{F}\), a reduction to \(10^\circ\text{F}\) 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°F to 10°F; this produced a further cost saving of around 8%. Thus a value of 10°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.
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 \(C_2\)\(^s\) 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 first section is comprised of the deethanizer, depropanizer and C\textsubscript{3} splitter and covers a temperature range of 160°F down to -64°F. There are no streams suitable for recovery of cold by sale to the refrigeration unit.

The second, low temperature, section consists of the demethanizer and high and low pressure C\textsubscript{2} splitters and covers a temperature range of -64°F down to -210°F. 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**

<table>
<thead>
<tr>
<th>Column Conditions</th>
<th>Pressure (psia)</th>
<th>( \frac{R}{R_{\text{Min}}} )</th>
<th>Key Splits (Mole Fractions)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Keys Overhead Bottom</td>
</tr>
<tr>
<td>Colunn</td>
<td></td>
<td></td>
<td>C(_2)O (0.16 ) (0.003)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C(_3) (0.015 ) (0.57)</td>
</tr>
<tr>
<td>Deethanizer</td>
<td>250</td>
<td>1.2</td>
<td>C(_1) (0.98 ) (0.005)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C(_2) (0.01 ) (0.58)</td>
</tr>
<tr>
<td>Demethanizer</td>
<td>75</td>
<td>1.1</td>
<td>C(_2) (0.96 ) (0.03)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C(_2) (0.45 ) (0.54)</td>
</tr>
<tr>
<td>High Pressure C(_2) Splitter</td>
<td>65</td>
<td>1.1</td>
<td>C(_2) (0.96 ) (0.04)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C(_2) (0.01 ) (0.94)</td>
</tr>
<tr>
<td>Low Pressure C(_2) Splitter*</td>
<td>20</td>
<td>1.1</td>
<td>C(_3) (0.25 ) (0.04)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C(_4) (0.04 ) (0.84)</td>
</tr>
<tr>
<td>Depropanizer</td>
<td>200</td>
<td>1.2</td>
<td>C(_3) (0.90 ) (0.08)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C(_3) (0.08 ) (0.76)</td>
</tr>
</tbody>
</table>

**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 C2 splitter (it is actually colder than any remaining cold stream) there remain 5 process streams within a temperature range of -66°F to -117°F, a difference of only 51°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 $17.3 \times 10^{-5}$, cf. $25.0 \times 10^{-5}$ 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°F to 7°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 $-66^\circ$F to $-190^\circ$F. It became obvious that unless either of cold streams -1, at $-87^\circ$F, or -2, at $-81^\circ$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
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
FIGURE 19. MODIFIED LOW TEMPERATURE SECTION
Table 8

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

<table>
<thead>
<tr>
<th>ITEM</th>
<th>PROCESS</th>
<th>High Pressure</th>
<th>Low Pressure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>#1</td>
<td>#2</td>
</tr>
<tr>
<td>Process Section Costs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process Network</td>
<td></td>
<td>734,000</td>
<td>568,000</td>
</tr>
<tr>
<td>(Including Columns)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Refrigeration Unit</td>
<td></td>
<td>476,000</td>
<td>948,000</td>
</tr>
<tr>
<td>Equipment Cost Breakdowns</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Columns</td>
<td></td>
<td>175,000</td>
<td>156,000</td>
</tr>
<tr>
<td>Heat Exchangers</td>
<td></td>
<td>231,000</td>
<td>322,000</td>
</tr>
<tr>
<td>Compressors</td>
<td></td>
<td>511,000</td>
<td>647,000</td>
</tr>
<tr>
<td>Service Costs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Steam</td>
<td></td>
<td>57,500</td>
<td>67,500</td>
</tr>
<tr>
<td>Water</td>
<td></td>
<td>22,200</td>
<td>37,200</td>
</tr>
<tr>
<td>Power</td>
<td></td>
<td>213,300</td>
<td>287,300</td>
</tr>
<tr>
<td>Other</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ethylene Product (lb moles/hr)</td>
<td>302.4</td>
<td>309.4</td>
<td>309.4</td>
</tr>
<tr>
<td>Ethylene Saving ($/yr)</td>
<td></td>
<td>-</td>
<td>48,000</td>
</tr>
<tr>
<td>Total Compressor HP</td>
<td></td>
<td>5,090</td>
<td>6,903</td>
</tr>
<tr>
<td>Power (kWh/Ton)</td>
<td></td>
<td>897</td>
<td>1190</td>
</tr>
<tr>
<td>TOTAL COST/YR</td>
<td></td>
<td>1,210,000</td>
<td>1,469,000</td>
</tr>
</tbody>
</table>
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\textsuperscript{(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°F to +200°F and 1 atm to 40 atm 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 convergence 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) Convergence 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 (\(\delta\)) 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 Sirola 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 foresee 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 inter-
polation.
iv) A systematic price \( \delta \) 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 \( \delta \) function (refer to Figure 13) makes any guarantee of optimality difficult, since the use of any finite step in \( \delta \) makes it possible that solutions may be missed. However for practical purposes the following algorithm is suggested for the general multi-dimensional \( \delta \) case. It can be argued intuitively and from the present computational results that the cost function (of \( \delta \)) is unimodal. Thus it is suggested that the practical \( \delta \) space (say from \(-5\) to \(+10\) times \( \Delta T_{\text{min}} \)) 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\(^\text{(37)}\). 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\(^\text{(6)}\) 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:


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


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


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

\[
P = \int_{T_1}^{T_2} pr(\theta \pm \delta) \frac{dH}{d\theta} d\theta
\]  

(I.1)

The function \( pr \) is defined by a known cubic spline in \( T \) and the value and sign of the \( \delta \) 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

\[
P_{AB} = \frac{\Delta H_{AB}}{\Delta T_{AB}} \int_{T_A}^{T_B} pr(\theta \pm \delta) d\theta
\]  

(I.2)

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

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

(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 \( \Delta Q \) is given by:

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

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

\[
\frac{\Delta S}{\Delta Q} = \frac{-1}{T_M + \frac{a}{2}} + \frac{1}{T_M - \frac{a}{2}}
\]

which simplifies to

\[
\frac{\Delta S}{\Delta Q} = \frac{a}{T_M^2 - \frac{a^2}{4}}
\]

and since the temperature difference, \( a \), will be much smaller than the absolute temperature \( T_M \), then
\[
\frac{\Delta S}{\Delta Q} = \frac{a}{T^2 M} \quad \text{(I.7)}
\]

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
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.

STMOV C is the stream handling utility routine which moves stream properties information between the stream matrices and working vectors.
PROGRAM MAIN
INPUT=1001, OUTPUT=1001, PUNCH=1001, TAPE5=INPUT, TAPE6=OUTPUT, TAPE7=PUNCH

*** COMMON DECK ***

DUMMY ARRAYS ARE FOR NAMELIST INPUT

COMMON / CONTL / NE, NIN, NOUT, NOCOMP
COMMON / KPM / N1S, NKPM, KPM1(6), KPM2(6), KPM3(6), KPM4(6), KPM5(6), KPM6(6), KPM7(6), KPM8(6), KPM9(6), KPM10(6),
1 KPM11(6), KPM12(6), KPM13(6), KPM14(6), KPM15(6), KPM16(6), KPM17(6), KPM18(6), KPM19(6), KPM20(6), KPM21(6), KPM22(6), KPM23(6), KPM24(8), KPM25(18),
4 SMPB1(15), SMPB2(15), SMPB3(15), SMPB4(15), SMPB5(15), SMPB6(15), SMPB7(15), SMPB8(15), SMPB9(15), SMPB10(15), SMPB11(15), SMPB12(15), SMPB13(15), SMPB14(15), SMPB15(15), SMPB16(15), SMPB17(15), 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 / DPIN(4), DPIN(4), DIN(4), HOUT(4), VFIN(4), TMIN(4),
1 XIN(8, 4)
COMMON / SOUT / POUT(4), POUT(4), TOUT(4), HOUT(4), POUT(4), VFIN(4),
1 TMOUT(4), XOUT(18, 4)
COMMON / PROP / COMPN (8), APC(8), ATC(8), AVC(8), AMW(8), AQMEG(8),
1 ADEL(8), AWI(8), APH(8), BET(8), GAM(8), ETA(8), SBASE(8), SBASE(8),
2 ZCD(8), ALD(8)
COMMON / AMORT / HRS, TSWAT, DTW, CWAT, TS, HVS, CS, CKWH, APPP, APRR,
1 ARR, TRRR

***

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)

NAMELIST / KPMLIST / N1S, NKPM, KPM1, KPM2, KPM3, KPM4, KPM5, KPM6, KPM7, KPM8,
1 KPM9, KPM10
NAMELIST / SMPALST / 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,
4 SMPB11, SMPB12, SMPB13, SMPB14, SMPB15, SMPB16, SMPB17, SMPB18, SMPB19,
8 SMPB20, SMPB21, SMPB22, SMPB23, SMPB24, SMPB25
NAMELIST / EMIILST / EMI1, EMI2, EMI3, EMI4, EMI5, EMI6, EMI7, EMI8, EMI9, EMI10
NAMELIST / PARLIST / AMORT, HRS, TSWAT, DTW, CWAT, TS, HVS, CS, CKWH, APPP, APRR,
1 ARR, TRRR
NAMELIST / COMP / NOCOMP, COMPNT

PRE-ZERO ARRAYS
CALL ZERO1 (KPM, 60)
CALL ZERO (SMPA, 725)

READ COMPONENT INFORMATION
READ (5, COMP)
READ COMPONENT PHYSICAL CONSTANTS
DO 10 I = 1, NOCOMP
10 READ (5, 11) (PROP(I, K), K = 1, 15)

READ TITLE
READ (5, 100) TITLE
IF (EOF, 11, 12)
1 CALL EXIT
2 WRITE (6, 101) TITLE
READ GENERAL SYSTEM PARAMETERS
READ (5, PARLIST)
READ PROCESS MATRIX
READ (5, KPMLST)
READ STREAM MATRICES (A+B SECTIONS)
READ (5, SMPALST)
READ EQUIPMENT MATRIX
READ (5, EMIILST)
C CALCULATE INPUT STREAM BUBBLE, DEW POINTS + ENTHALPIES
DO 20 J=1,NIS
IF(SMP8(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.)
SMPR(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

COMPUTES COLUMN SYSTEM, IDENTIFIES HEAT TRANSFER + EXPANSION/COMPRESSION TASKS AND MAKES ENTRIES IN STREAM SPEC VECTORS

KPM PROCESS MATRIX CODING ••
FOR EACH (EQUIPMENT) VECTOR OF KPM -
1. EQUIPMENT NUMBER
2. EQUIPMENT TYPE NUMBER
3. -6. INPUT , THEN OUTPUT STREAM NUMBERS (+ INPUT, - OUTPUT)

SMPA STREAM CONTROL VECTOR CODING -
1. STREAM NO
2. STREAM TYPE - 1. FEED, 0. INTERMEDIATE, -1. PRODUCT
3. FURTHER PROCESSING FLAG - 1. FOR FURTHER PROCESSING
4. PRESSURE SPEC
5. PHASE SPEC (1. + VAPOR FRACTION)
6. TEMPERATURE SPEC

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,SWAT,DTW,CTH,TS,HVS,CS,CKWH,APPR,APPR,
1 ARR,RTRR

DIMENSION DMCHA(8),DUM(8),NSCH(10,2),FSAV(4,2),NNPR(2)
INITIALIZE MATRIX COUNTERS
NNPR(2)=NNPR(1)=0
NSM=NIS

DO 6 NE=1,NKPM
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

SCAN EQUIP STREAMS FOR DEMANDS
DO 8 JS=JF,JL
NS=JIO*KPM(1,JS)+NE
NF=JS-2
IF(SMPA(3,NS),EQ,0.) GO TO 12
12 JTYPE=SMPA(2,NS)
FOR INTERMEDIATE STREAMS (JTYPE=0) PROCESS ONCE ONLY - AS EQUIP ••
INPUTS - I.E. DON'T PROCESS AS EQUIP OUTPUTS
IF(JTYPE.EQ.0.)AND.(JIO.EQ.-1) GO TO 12
PRESS=SMPA(4,NS)
PHAS=SMPA(5,NS)
TEMP=SMPA(6,NS)

SATISFY DEMANDS - MOVE STREAM INTO SIN(1) FOR PROCESSING
CALL MVFSM(1,NS,0,0)
ASSIGN 12 TO LOC
GO TO 100
8 CONTINUE
IF(JIO.EQ.-1) GO TO 35

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

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)DPIN(1),DPIN(1),TIN(1),PIN(1),HIN(1),VFIN(1),TMIN(1),
14(XIN(I,1),I=1,NOCOMP)
COMPUTE EQUIP (COLUMN)
CALL DIST
PCOND=EQUIP(4)
TCOND=EQUIP(5)
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)DPOUT(J),DPOUT(J),TOUT(J),POUT(J),HOUT(J),VFOUT(J),
1TOUT(J),(XOUT(I,J),I=1,NOCOMP)
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)
JL=JF+NO-1
SCAN OUTPUTS FOR DEMANDS FOR COLUMNS THERE ARE TWO SPECIAL OUTPUTS -
SOUT(3) TO BE CONDENSED (OR P.C.)
SOUT(4) TO BE REBOILED
SCAN NORMAL OUTPUTS
J10=-1
GO TO 8
SATISFY DEMANDS FOR SPECIAL OUTPUTS
35 JTYPE=0
DO 45 J=3,4
TEMP=PRES=PHAS=0.
MOVE INTO SIN(1) FOR PROCESSING
CALL MVSO(1,J,0,0)
IF(J,J.EQ.4) GO TO 38
OVERHEADS VAPOR FLOW TO CONDENSER (TOTAL OR PARTIAL)
IF(PCOND.EQ.0) PHAS=1.
IF(PCOND.EQ.1) TEMP=TCOND
GO TO 40
BOTTOMS LIQUID FLOW TO REBOILER
38 PHAS=2.
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
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

**DEMAND ROUTINE**

**OPERATES ON SIN(1)**

**IDENTIFY STREAMS AS HOT (ICH=1) OR COLD (ICH=2)**

**CONTINUE**

**IF** (TEMP.LT.TIN(1)) **ICH**=1

**GO TO 110**

**IF** (PHASE.LT.0.) **GO TO 102**

**IF** (PHASE.EQ.0.) **GO TO 106**

**IF** (PHASE.EQ.1.) **GO TO 104**

**IF** (RES.EQ.PIN(1)) **SMPA**(4,NS)=0.

**PRESSURE**

**IF** (PRES.EQ.0.) **PRES**=PIN(1)

**IF** (PRES.EQ.PIN(1)) **SMPA**(4,NS)=0.

**PHASE**

**TEMP SPEC** OVERRIDES PHASE SPEC

**IF** (PHASE.EQ.1.) **PHASE**=VFIN(1)

**IF** (TEMP.GT.0.) **GO TO 120**

**IF** (PHASE.EQ.VFIN(1)) **GO TO 120**

**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

**TEMP**

**IF** (TEMP.LE.0.) **TEMP**=TIN(1)

**IF** (TEMP.EQ.TIN(1)) **GO TO 130**

**SET FLAG FOR FINDING VF**

**IVF**=1

**SET POST-PROCESSED STREAM PROPS - FOR EQUIP INPUTS ONLY**

**IF** (J10.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 MVSO(1,1,0,0)**

**GO TO 145**

**VF**=PHASE

**CALL ENTH(1,HIN(1),DUM)**

**STORE POST-PROCESSED STREAM PROPS**

**FSAV**(1,NF)=TIN(1)

**FSAV**(2,NF)=PIN(1)

**FSAV**(3,NF)=HIN(1)

**FSAV**(4,NF)=VFIN(1)

**GO TO LOC,(12,45)**

**FORMAT(215)**

**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 STMOVCIWV(ISM,IVW,III,NX)

STREAM MOVING UTILITY ROUTINE ...(COLSYS VERSION)

IVW  - ELEMENT NUMBER IN SIN OR SOUT WORKING ARRAY
  + SIN
  - SOUT

ISM  - VECTOR NUMBER IN SM--

III  - 0  MOVE TO OR FROM SMPB
       1-2  MOVE TO OR FROM SMCHB - (1-2)
       3-5  MOVE TO OR FROM SMRB  - (1-3)

NX  - VECTOR NUMBER CONTAINING MOLE FRACTIONS (III GT 0)

3 ENTRIES -
1 MVSO SI MOVES SOUT VECTOR ISM TO SIN VECTOR IVW (III=0)
2 MVFSM MOVES FROM SM-- TO SIN OR SOUT
3 MVTSM MOVES TO SM-- FROM SIN OR SOUT

*** 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(4,8)
COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
1 TMOUT(4),XOUT(8,4)

***
DIMENSION SMCHB(1,1,1),SMCHX(1,1,1),SMRB(1,1,1),SRMX(1,1,1)
DIMENSION SIDUM(4,7),SODUM(4,7)
EQUIVALENCE (BPIN,SIDUM),(BPOUT,SODUM)

C**
ENTRY MVSO SI
IEN T=1
GO TO 1
C**
ENTRY MVFSM
IEN T=2
GO TO 1
C**
ENTRY MVTSM
IEN T=3

1 JJJ=III+1
GO TO (2,3,4,4)JJJ
2 ITYPE=1
GO TO 5
3 ITYPE=2
KKK=III
GO TO 5
4 ITYPE=3
KKK=III-2

5 GO TO (100,200,300)IEN T
C**
MVSO SI
100 DO 50 I=1,7
50 SIDUM(IWV,I)=SODUM(ISM,I)
DO 60 I=1,NOCOMP
60 XIN(I,IWV)=XOUT(I,IS M)
RETURN
C**
MVFSM
100 DO 50 I=1,7
50 SIDUM(IWV,I)=SODUM(ISM,I)
DO 60 I=1,NOCOMP
60 XIN(I,IWV)=XOUT(I,IS M)
RETURN
C*
GO TO 18
AA=SMCHX(I,NX,KKK)*SMCHB(7,ISM,KKK)
GO TO 18
AA=SMRX(I,NX,KKK)*SMRB(7,ISM,KKK)
IF(IWV.GT.0) XIN(I,IWV)=AA
IF(IWV.LT.0) XOUT(I,-IWV)=AA
20 CONTINUE
RETURN
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
IF(I$type.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.

SIMOVs 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.
DO 50 ALL STREAM PRESSURE SPECIFICATIONS

SELECT APPROPRIATE EQUIPMENT

110
DO 200/201 STREAM MATCHING FOR TEMPERATURE SPECIFICATIONS

PRE-SCREEN BY HEURISTICS IS MATCH "TYPE INFEASIBLE"?

50

110

DO 200/201 STREAM MATCHING FOR TEMPERATURE SPECIFICATIONS

SELECT APPROPRIATE EQUIPMENT

400

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

```
PROGRAM MAINS (INPUT=1001, OUTPUT=1001, DUMMY=1001, TAPE5=INPUT, TAPE6=OUTPUT, TAPE8=DUMMY)

*** COMMON DECK ***

COMMON/FEAT/IFTR(10)
COMMON/CNTL/NE,NIN,NOUT,NOCOMP
COMMON/PROP/COMPNT(8),APC(8),ATC(8),AVC(8),AMW(8),AOMEG(8),
  ADEL(8),AVW(8),APH(8),BET(8),GAM(8),DTA(8),BASEA(8),BASEB(8),
1 ZCD(8),ALD(8)
COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APP,APR,
A RRR,IRR
COMMON/PATH/JPATH(6,200),NPATH(20),NPHTS
COMMON/REFL/NLEV,RLEV(10)

** 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)

***

INTEGER COMPNT
DIMENSION TITLE(8),PROP(8,15),XX(8)
EQUIVALENCE (PROP,APC)

NAMELIST/PARLIST/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APP,APR,
1 A RRR,IRR
NAMELIST/COMP/NOCOMP,COMPNT

READ COMPONENT INFORMATION
READ(5,COMP)
READ COMPONENT PHYSICAL CONSTANTS
DO 10 I=1,NOCOMP
10 READ(5,11)(PROP(I,K),K=1,15)

READ TITLE
READ(5,100)TITLE
IF (.EOF(5)),1,2
1 CALL EXIT
2 WRITE(6,101)TITLE
READ FEATURE CARD
READ(5,105)IFTR
READ GENERAL SYSTEM PARAMETERS
READ(5,PARLIST)

READ SMCH A+B
READ(5,105)NNPR,NNSER
DO 50 K=1,2
      NNN=NNSM(K)=NNPR(K)
DO 30 J=1,NNN
30 WRITE(6,31)(SMCHA(L,J,K),L=1,8)

DO 40 J=1,NNN
40 WRITE(6,32)(SMCHB(L,J,K),L=1,8)

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

READ REFRIGERATION LEVELS
READ(5,105)NLEV
READ(5,33)(RLEV(I),I=1,NLEV)
CALL SMATCH

11 FORMAT(3(/5E14.5))
31 FORMAT(5F6.2,5X,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(105)
END
```
SUBROUTINE SMATCH

COMPUTES ALL EQUIPMENT SEQUENCES TO SATISFY TEMPERATURE + PRESSURE DEMANDS FOR A SET OF HOT + COLD STREAMS (ICH=1 HOT + 2 COLD)

1. SATISFIES ALL PRESSURE CHANGE DEMANDS BY ADIABATIC EXPANSION OR MULTISTAGE COMPRESSION (VAPORE ONLY)
2. COMPUTES ALL POSSIBLE MATCHES BETWEEN STREAMS, RESIDUALS + SERVICE STREAMS (STEAM, WATER, REFR)

FEATURES ARE ACTIVATED BY 1 ENTRY IN /FEAT/
1. STEAM
2. SALE
3. WATER
4. REFRIGERATION
5. VAPOR RECOMPRESSION

PRE-ASSIGNED RULES ARE USED TO PRE- SCREEN POSSIBLE MATCHES

SMCHA - STREAM CONTROL VECTORS -
1. PRIMARY STREAM NO
2. SECONDARY STREAM NO
3. ACTIVE/INACTIVE FLAG - 0. ACTIVE, 1. INACTIVE
4. STREAM TYPE - 1. FEED, 0. INTERMEDIATE, -1. PRODUCT (2. HIGH PRIORITY - SATISFY BY SERVICE STREAM ONLY)
5. PSEUDO-SERVICE STREAM - PARALLEL PROCESSING
6. FREE
7. PRESSURE SPEC
8. TEMP SPEC (-1. = FREE)

MATCH COUNTER ARRAYS - MACC, MREJ
MACC - ACCEPTED
1. STEAM
2. SALES
3. WATER
4. REFR
5. PROCESS STREAM
6. ADIABATIC EXPANSION
7. COMPRESSION

MREJ - REJECTED - (2-5 APPLY ONLY TO PROCESS/PROCESS MATCHES)
1. GENERAL TECH INFEASIBILITY
2. VAP RECOMP - VRTMX EXCEEDED
3. EXCEEDS MAX ENTROPY INCREASE/BTU (DENMX)
4. VAP/VAPORE EXCEEDS MAX INLET TEMP DIFF (SUSP FOR HOT SOURCE)
5. Q LT QMIN
6. VAPOR/VAPOR MATCH
7. FEED/INTERMEDIATE MATCH

*** COMMON
COMMON/FEAT/IFTR(10)
COMMON/CONT/L/E/N,IN/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/BOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
1 TMOUT(4),XOUT(8,4)
COMMON/PARAM/AMORT,HRS,TWAT,DWAT,CWAT,TS,HVS,CS,CKWH,APPF,APRE,
1 ARR,1RRR
COMMON/PATH/JPATH(8,200),NPATH(20),NPETHS
COMMON/REF/R/NLEV,RLEV(10)
BLANK COMMON
COMMON/NPR(2),NNSER(2),NNSM(2),SMCHA(8,50,2),SMCHB(7,50,2),
1 SMCH(8,10,2)
COMMON/NEMC,EMCH(15,100)

** DIMENSION JIN(2),ICH(2)
DIMENSION JCHA(2,6),NPR(2),NSS(2),JACT(2),JTP(2),JSTP(2)
DIMENSION PSPEC(2),TPEC(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))

DATA PRMAX,RDMAX,VRTMX/4.,50.,60.
DATA QMIN,TLIM/2.5E5,470./
DATA DENMX,DMAX/25.5,100./
C INDEX DISPLACEMENT FUNCS FOR JPATH + NPATH ARRAYS
DATA NPHTHS/20/
IDJ(JPR,JCH)=NPHTHS*((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*** SATISFY ALL PRESSURE SPECIFICATIONS ***
WRITE(6,771)
IPRES=1
C DO 50 KCH=1,2
NNS=NSM(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
IIF(PRES.NE.PIN(1)) SMCHA(3,JS,KCH)=1.
IF((PRES-PIN(1)) 15,50,20
C** ADBIATIC FLASH
C CALL ADBF, LOAD EQUIP + OUTPUT STREAM
15 MACC=7
ACTF=0.
IOP=10
JIN(2)=0
ASSIGN 50 TO LOADSE
GO TO 400
C** COMPRESSION
C GO TO COMPRESSION ROUTINE
20 GO TO 450
C 50 CONTINUE
WRITE(6,772)
GO TO 70
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 MVS(MI,JS,KCH,KPR)
GO TO JLD,(10,115)
C***********
C*** COMPUTE ALL POSSIBLE HEAT EXCHANGE MATCHES ***
C 70 IPRES=0
C SET UP COUNTERS - SCAN ACROSS C FOR EACH H
MIN1=-1
NHD2=NC2=0
110 NNH2=NSM(1)
NNC2=NSM(2)
C DO 201 NNH=MIN1,NNH2
DO 200 NNC=MIN1,NNC2
C HAS MATCH BEEN COMPUTED BEFORE
IF(NNH.LE.NHO2.AND.NNC.LE.NCO2) GO TO 200

ISERV=0
ACTF=0.

LOAD STREAM CONTROL INFO INTO WORKING VECTORS
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

TEST CONTROL INFORMATION
115 IF(JACT(NI).EQ.1) GO TO 200
IF(JTPT(NI).EQ.-2) ISERV=NI
120 CONTINUE

--- SPECIFIC MATCH SELECTIONS/REJECTIONS ---
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

*HEURISTIC* PRE-SCREENING TO REJECT TYPE-INFEASIBLE MATCHES
122 IF(JSTP(1).NE.0.AND.JSTP(2).NE.1) GO TO 200
123 FOR JTP(J).EQ.1.AND.JTP(3-J).EQ.0) GO TO 212
FOR JTP(1).NE.0.AND.JNC.GT.0) GO TO 200
IF(JTPT(2).EQ.2.AND.NNH.LT.-1) GO TO 200
GO TO 156

COLD STREAM *NNC* - MATCHES WITH STEAM(NNH=-1), VALUE(NNH=0)

125 IF(NNH.EQ.0) GO TO 130
IF(FTR(1).EQ.0) GO TO 200
IF(JSTP(1).EQ.1) GO TO 200
CALL HXER + LOAD EQUIP
NACC=1
IOP=2
IS2=1
ASSIGN 200 TO LOADSE
GO TO 400

SALE - VALID FOR SOURCE ONLY
130 IF(FTR(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

HOT STREAM *NNH* - MATCHES WITH WATER(NNC=-1), REFR(NNC=0)
- INVALID FOR SOURCE

140 IF(JSTP(1).EQ.1.AND.NPR(1).NE.1) GO TO 200
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 TWA+APPR
TEX(1)=TWA
CALL HXER + LOAD EQUIP
NACC=3
ASSIGN 200 TO LOADSE
GO TO 400
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 J=1,NLEV
152 IF(ABS(TEX(1)-RLEV(1)).LT.10.) TEX(1)=RLEV(1)+APPR
C MAKE EQUIP ENTRY + LOAD RESID IF NECC
154 NACC=4
ASSIGN 200 TO LOADSE
GO TO 400
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* TEST INLET TEMPS
162 APMIN=APPR
IF(TIN(2).LT.TRRR) APMIN=APPR
DTIN=TIN(1)-TIN(2)
C CHECK FOR DTIN GT MAX (EXCEPT FOR HOT SOURCE)
C 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 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 COMPUTE CLOSEST APPROACH + CHECK WHETHER GT MIN
CLAP=AMIN(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) .AND. (3-J).EQ.TIN(J)-SIGN*APMIN .AND. IOP=1)
IS=4+ISERV
CALL HXER, LOAD EQUIP, TEST FOR RESIDUALS + LOAD OUTPUT STREAMS
ASSIGN 200 TO LOADSE
GO TO 400
C** VAPOR RECOMPRESS

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(NS5(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* REJECT IF PHASE CHANGES NOT POSSIBLE FOR BOTH STREAMS
DO 250 J=1,2
TI=TIN(J)
TX=TSPEC(J)
BP=BPINC(J)
DP=DPINC(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* APPROX TEMP DIFFERENCE TEST
IF(MAX(ABS(DT(J)),ABS(DT(J))) .LT. VRTMX) GO TO 204
C* TEST FOR PRESSURE RATIO TO ACHIEVE MIN APPROACH AT COLD EXIT
TSAV=TIN(1)
PSAVE=PIN(1)
TIN(1)=TSPEC(2)+APRR
PIN(1)=2.*PIN(1)
CALL PDEW(1,PRES,ST)
WRITE(6,766)NNH,NNC,PSAVE,PRES
TIN(1)=TSAV
PIN(1)=PSAVE
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.
C* GO TO COMPRESSION ROUTINE
GO TO 450
C SET NEW TEMP SPEC FOR OUTLET STREAM
172 TSPEC(1)=TEX(1)=BPINC(1)
JINC(2)=NNC
C* EXCHANGE
GO TO 168
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)JINC,NREJ
200 CONTINUE
201 CONTINUE
C COMPUTE NO OF NEW RESIDUALS

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
NH02=NNH2
GO TO 110

C SAVE OLD COUNTERS, SET NEW COUNTERS + COMPUTE FOR NEW RESIDUALS

NCO2=NNC2
NH02=NNH2
GO TO 110

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

300 STR=(JN*(3-2*JCH))
NCJ=IDJ(JPR,JCH)
NROW=JSS+1
IF(IPRES.EQ.0) GO TO 308

C MAKE ENTRY IN COL 1
NCOL=NCJ+1
NROW=JPATH(1,NCOL)+1
GO TO 330

C IS STREAM NON-PRIMARY

308 IF(NROW.NE.1) GO TO 310
C CREATE NEW COL
JC=2
GO TO 315
LOCATE EQUIP FOR WHICH STREAM STR IS AN * OUTPUT *
SCAN ROW NROW
310 DO 312 JC=2,NPTHS
NEQ=JPATH(NROW,JC+NCJ)
IF(NEQ.EQ.0) GO TO 312
IF(EMCH(6,NEQ).EQ.STR.OR.EMCH(6,NEQ).EQ.STR) GO TO 314
312 CONTINUE
314 NCOL=NCL=JC+NCJ
IS NEXT ENTRY IN COL FREE
IF(JPATH(NROW+1,NCOL).EQ.0) GO TO 330
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)
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
********************************************************************
*** EQUIP CALLING ROUTINE **
INCREMENT MATCH COUNTERS
400 NACC=NACC+1
JIN=2
IF(JIN(2).LE.0) JIN=1
WRITE(6,700)JIN,IOP
DO 706 J=1,JIN
706 WRITE(6,622)(JCHAI(J;II),II=1,5),PSPEC(J),TSPEC(J)
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
ADIABATIC FLASH **
401 NOUT=NIN+1
CALL ADBF(PRES)
GO TO 500
COMPRESSOR **
410 CALL COMP(PRES)
GO TO 500
HEAT EXCHANGER **
420 IF(IS2.LE.3) JIN(2)=IS2+200
IF(IS2.LE.3) GO TO 480
CALL HXER(IOP,IS2,TEX,Q)
IF(IS2.LE.4) GO TO 500
IF(ARS(Q).LT.QMIN) GO TO 212
ENTROPY INCREASE TEST FOR PROC/PROC MATCH (NOT FOR HOT SOURCE)
C* IF(EQUIP9).LT.DENMX.OR.JSTP(1).EQ.1) GO TO 500
WRITE(6,778)EQUIP9
GO TO 206
MULTISTAGE) COMPRESSION ROUTINE - WATER INTERCOOLING IF REQD
C COMPUTE NO OF STAGES - EQUAL PRES RATIO/STAGE
450 PR=PRES/PIN(1)
STAGE=ALOG(PR)/ALOG(PMAX)
NSTG = INT(STAGE + 0.999)
PRSTG = PR** (1. / FLOAT(NSTG))

DO 460 KK=1,NSTG
ACTF = 1.
PRES = PIN(1) * PRSTG
CALL COMPR, LOAD EQUIP + OUTPUT STREAM
NACC = 8
IOP = 1
JIN(2) = 0
ASSIGN 452 TO LOADSE
GO TO 400

452 JIN(1) = NS
NO AFTERCOOLING FOR IPRES = 0 , KK = NSTG
IF(KK * EQ * NSTG AND IPRES .EQ. 0) GO TO 460
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*** EQUIP ENTRY ONLY - UNLESS RESIDUAL IS INDICATED
EQUIP TYPES - (1. REF) , (30. SALE)

480 CALL ZERO(EQUIP, 15)
EQUIP(2) = IOP
IF(IOP .EQ. 3U) GO TO 482
EQUIP(10) = TOUT(1) = TIN(1) = TEX(1)
COMPUTE OUTLET STREAM CONDITION IF TSPEC NOT MET
NIN = NOUT = 1
IF(TEX(1) .NE. TSPEC(1)) CALL ISOFIO.
GO TO 500

482 EQUIP(7) = TIN(1)
EQUIP(8) = TEX(1)
GO TO 500

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(JIN * EQ. 2) EQUIP(4) = JIN(2) * (-ISIGN1)
WRITE(6, 662)
IF(IPRES .EQ. 1) GO TO 502
FOR PSEUDO-SERVICE STREAM ADD 300. TO STREAM CODE
IF(ISERV .GT. 0) EQUIP(ISERV + 2) = EQUIP(ISERV + 2) + 300.
C*
MAKE ENTRIES IN STREAM PATH ARRAY - FOR INPUT STREAMS
- 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*
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* 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
C INCR SECONDARY STREAM NO NSS(NO)=NSS(NO)+1
C* SET REMAINING CONTROL INFORMATION FOR NEW STREAM C++ IF(ACT*INT*,LOAD SET TYPE TO 2)
C* IF(ACTF.EQ.0..AND..JTP(NO).EQ.0..AND..JSTP(NO).EQ.0) JTP(NO)=2
C* 520 JACT(NO)=ACTF
C 522 ACTF=0.
DO 522 K=1,6
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(1,1,1,0)
WRITE(6,750)
WRITE(6,622)(SMCHA(I,NS,JCH),I=1,8)
WRITE(6,624)(SMCHA(I,NS,JCH),I=1,7)
EQUIP(NO+4)=NS*(3-2*JCH)
530 CONTINUE
C* LOAD EQUIP INFORMATION DO 542 K=1,16
EQUIP=6,770)EQUIP
GO TO LOADSE,(50,200,452,460)
C 600 FORMAT(1015)
612 FORMAT(7F10.4)
622 FORMAT(5F8.0,5X,3F8.1,17)
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(I/)
700 FORMAT(/1H.12(1H*),/2H.2I4.3H.15/I1H.12(1H*))
750 FORMAT(/..*++ LOADED OUTPUT-*)
760 FORMAT(/..-- VAP RECOMP - NNH,NNC*,215,* 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,14)
774 FORMAT(/..* MACC*,1015/* MREJ*,1015/* NSMH,NSMC,NRSH,NRSC*,
775,10X,2I5/* NFMCH*,15/H1)
776 FORMAT(* Nay*,NRow*COL*,216,5X,615)
777 FORMAT(/1H.2W1(IH*),* NPHS EXCEEDED , JPR,JCH=*,215)
778 FORMAT(* DENT*,F8.1)
END
SUBROUTINE STMOVS(IWV, ISM, III, NX)

STREAM MOVING UTILITY ROUTINE ...(SMATCH + BRBND VERSION)

IWV - ELEMENT NUMBER IN SIN OR SOUT WORKING ARRAY
  + SIN
  - SOUT
ISM - VECTOR NUMBER IN SM--
III - 0 MOVE TO OR FROM SMPB
      1-2 MOVE TO OR FROM SMCHB - (1-2)
      3-5 MOVE TO OR FROM SMRB - (1-3)
NX - VECTOR NUMBER CONTAINING MOLE FRACTIONS (III GT 0)

3 ENTRIES -
1 MVSSM MOVE SOUT VECTOR ISM TO SIN VECTOR IWV (III=0)
2 MVFSM MOVES FROM SM-- TO SIN OR SOUT
3 MVTSM MOVES TO SM-- FROM SIN OR SOUT

*** COMMON DECK ***
COMMON/CONTLYNE,NIN,NOUT,NOCOMP
COMMON/SIN/BBIN(4),DPIN(4),TIIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
  1 XIN(8,4)
COMMON/SOUT/BPOUT(4),DPOUT(4),TOU4T(4),POUT(4),HOUT(4),VFOUT(4),
  1 TMOUT(4),XOUT(8,4)
C**
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**
ENTRY MVSSMI
IEN=1
GO TO 1
C**
ENTRY MVFSM
IFN=2
GO TO 1
C**
ENTRY MVTSM
IEN=3

1 JJJ=III+1
  GO TO (2,3,4,4,4)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)IEN

C**
MVSSMI
100 DO 5U I=1,7
  50 SIDUM(IWV,I)=SODUM(ISM,I)
  DO 5U I=1,NOCOMP
  60 XIN(I,1WV)=XOUT(I,ISM)
  RETURN

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
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*** MVTSN
300 DO 30 I=1,7
IF(IWV.GT.0) AA=SIDUM(IWV,I)
IF(IWV.LT.0) AA=SDUM(-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
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)
Routines generates stream history for stream JSS
- Compiles list of streams (JHIST, NO. NHIST) used in producing JSS

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)

DIMENSION JHIST(1),JIS(10)

INDEX DISPLACEMENT FUNCS FOR JPATH, NPATH ARRAYS
IDJ(JPR,JCH)=NPTH*S*((JPR-1)+NNPR(1)*(JCH-1))
IDNJPR(JCH)=JPR+NNPR(1)*(JCH-1)

JJ=NIS=0
NHIST=1
JHIST(1)=JSS
CALL ZEROI(JIS,10)
JSTR=JSS
GO TO 12

SELECT NEXT STREAM FROM JIS
10 JJ=JJ+1
IF(JJ,GT,NIS) RETURN
JSTR=JIS(JJ)
12 JCH=(J-JSIGN(1,JSTR))/2
JPR=ABS(JSTR)
JPR=SMCHA(1,JPR,JCH)
JSEC=SMCHA(2,JPR,JCH)
IF(NHIST.EQ.1.AND.JSEC.EQ.0) RETURN
IF(JSEC.EQ.0) GO TO 10

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

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

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
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
26 CONTINUE
30 CONTINUE
GO TO 10
END
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 incompatibility with bounding problems or because they must lead to networks of higher cost than 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 reactivate 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 BREND Algorithm (Continued on page 148 ).
Figure II.2 BRBND Algorithm (Continued).
** Program Mainb**

**COMMON DECK**

COMMON/CONT,NE,NIN,NOT,NOCOMP
COMMON/PROP/COMPNT(8),APC(8),ATC(8),AVC(8),AMW(8),AOMEG(8),
1 ADEL(8),AWW(8),APH(8),BET(8),GAM(8),DTA(8),BASEA(8),BASEB(8),
2 ZCD(8),ALD(8)
COMMON/PARAM/AMORT,HRS,TWAT,DTW,SWAT,TS,HVS,CS,CKWH,APPML4,APRR,
1 AARR,TRRR,DTF
COMMON/PLINE/NH,NC,X(10),Y(10),PM(10)
COMMON/PAT/H,JPATH(8,200),NPATH(20),NPATHS
BLANK COMMON
COMMON/NPR(2),NSER(2),NSM(2),SMCH(8,50,2),SMCHB(7,50,2),
1 SMCH(8,10,2)
COMMON/NEMCH,EMCH(15,100)

** INTEGER**

DIMENSION TITLE(8),PROP(8,15)
EQUIVALENCE (PROP,APC)
NAMENLIST/PCR/LST,AMORT,HRS,TWAT,DTW,SWAT,TS,HVS,CS,CKWH,APPML4,APRR,
1 AARR,TRRR,DTF
NAMENLIST/COMP/NOCOMP,COMPNT
INDEX DISPLACEMENT FUNCS FOR JPATH + NPATH ARRAYS
IDJ(JPR,JCH)=NPATH*((JPR-1)+NPR(1)*(JCH-1))
IND(JPR,JCH)=JPR+NPR(1)*(JCH-1)

READ COMPONENT INFORMATION
READ(8,COMP)
READ COMPONENT PHYSICAL CONSTANTS
DO 10 I=1,NOCOMP
10 READ(8,11)(PROP(I,K),K=1,15)

READ TITLE
READ(5,100)TITLE
IF(EOP,5,1,2)
1 CALL EXIT
2 WRITE(6,101)TITLE
READ GENERAL SYSTEM PARAMETERS
READ(5,PARLST)
READ BRANCH + BOUND PARAMETERS
READ(5,600),NPATHS,LBXX,NREJX,NMIN

READ STREAM,EQUIP + STREAM PATH INFORMATION
READ(8,600),NPR,NNSER,NSM
DO 625 JCH=1,2
610 NNN=NPR(JCH)
DO 610 J=1,NNN
620 NNN=NNSM(JCH)
READ(8,621)(SMCHX(I,J,JCH),I=1,NOCOMP)
CONTINUE
DO 625 JCH=1,2
JP1=IDJ(1,JCH)+1
JP2=IDJ(NPR(JCH),JCH)+NPATHS
NP1=IDN(1,JCH)
NP2=IDN(NPR(JCH),JCH)
READ(8,600)(NPATH(I),I=NP1,NP2)
DO 640 JP=JP1,JP2
640 CONTINUE
READ(8,600)(JPATH(I,K),I=1,8)

READ IN INITIAL REFERENCE + 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**

***** COMMON DECK *****

```
COMMON/CONTLINE,NIN,NOUT,NOCOMP
COMMON/PARAM,AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APP,AAPR,
  1 ARR,TRRR,DTF(2)
COMMON/EQUIP/EQUIP(15)
COMMON/SIN/DPIN(4),DPIN(4),TPH(4),PIN(4),HH(4),VF(4),TM(4),
  1 X(8,4)
COMMON/SPLINE/NH,NC,X(10),Y(10),PM(10)
COMMON/PATH/JPATH(8,200),NPATH(20),NPTH
COMMON/PLOPT/NEPT,NEOPT(40),NCOST(10)
BLANK COMMON
COMMON/NP(2),NNS(2),NS(2),SMCHA(8,50,2),SMCHB(7,50,2),
  1 SMCH(8,10,2)
COMMON/NEMCH,EMCH(15,100)
***
```

```
DIMENSION TEX(2)
DIMENSION SORT(20,2),KTAG(20),JPSV(8,20)
DIMENSION REFD(10,2),REFS(10,2),PSRV(5,2)
C INDEX DISPLACEMENT UNCS FOR JPATH + NPATH ARRAYS
ID(JPR,JCH)=NPATHS*[(JPR-1)+NP(1)*(JCH-1)]
IDN(JPR,JCH)=JPR+NP(1)*(JCH-1)
C JPATH ARRAY - EACH COL REPRESENTS ONE STREAM PROC PATH
  1. NO OF EQUIPS IN PATH
  2. -6. EQUIP NOS
  7. COST OF PATH
  8. ACTIVE(0)/INACTIVE(1) FLAG
C*****
```

**ENTRY ENEC**

```
** COMPUTES EQUIPS INVOLVING REFR + SALE (TYPES 1+30)
SUMS + SORTS COSTS FOR ALL STREAM PROCESSING PATHS
```

```
SCAN EMCH MATRIX
DO 50 KE=1,NEMCH
  IF(EMCH(2,KE).NE.30.) GO TO 10
  IOP=30
  GO TO 50
10 IF(EMCH(4,KE).NE.203.) GO TO 50
  IOP=1
  IS=3
C LOAD INPUT STREAM INTO SIN(1)
12 JS=EMCH(3,KE)
  JCH=(3-ISGN(1,JS))/2
  JN=ABS(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**
```

```
REF - FIND TEMP LEVEL LE TEX(2)
TEX(1)=EQUIP(10)
AP=AAPR
IF(TEX(1).LT.TRRR) AP=AAPR
T2=TEX(1)-AP
NCC=NC+1
DO 20 L=1,NCC
20 IF(X(L).LT.T2) GO TO 22
C COMPUTE EXCHANGER + SUM COSTS
CALL HXER(IOP,IS,TEX,Q)
EQUIP(10)=TEX(1)
EQUIP(14)=Y(LL)*HRS*ABS(Q)
EQUIP(15)=EQUIP(15)+EQUIP(14)
GO TO 40
C
```
** SALE - PUT TEMP DISPL=DTF*APRR**

** TEX=EQUIP(9)**

** IF(TEX.LE.0.) TEX=TWAT**

** DT=DTF(2)*APRR**

** CALL SVALUE(JCH,TEX,DT,VALUE)**

** LOAD EQUIP 7-15 INTO EMCH**

** DO 42 K=7,15**

** EMCH(K,KE)=EQUIP(K)**

** WRITE(6,444)(EMCH(K,KE),K=1,15)**

** CONTINUE**

** **************************

** SUM COSTS FOR EACH STREAM PROCESSING PATH IN JPATH ARRAY**

** FOR EACH PROCESS/PROCESS MATCH CHARGE HALF TO EACH STREAM**

** KEEP SEPARATE TOTAL FOR COL 1 (PRE-PROC) COSTS**

** WRITE(6,400)**

** COSTPP=0.**

** 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)**

** 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**

** NEO=JPATH(K,NCOL)**

** COST1=COST1+EMCH(15,NEQ) **

** COSTPP=COSTPP+COST1**

** 62 JPATH(7,NCOL)=COST1**

** SUM OTHER COLS + SORT IN ORDER OF INCR COST**

** NPS=JPATH(NCN)**

** NCJ=NCJ+1**

** DO 70 NC=1,NPS**

** NCOL=NC+NCJ**

** NNR=JPATH(1,NCOL)+1**

** COST=COST1**

** DO 66 K=2,NNR**

** NEO=JPATH(K,NCOL)**

** CST=EMCH(15,NEQ)/2.**

** IF(J2.EQ.0.OR.J2.GT.200) CST=CST*2.**

** 64 J2=EMCH(I,NEQ)**

** 66 COST=COST+CST**

** SORT(NC,1)=JPATH(7,NCOL)=COST**

** SAVE JPATH COLUMN**

** DO 68 NR=1,7**

** JPSV(NR,NC)=JPATH(NR,NCOL)**

** 70 CONTINUE**

** SORT + REPLACE IN ORDER**

** CALL TGSORT(SORT,KTAG,NPS,-1)**

** WRITE(6,401)**

** DO 74 NC=1,NPS**

** NCOL=NC+NCJ**

** NCC=KTAG(NC)**

** DO 72 NR=1,7**

** JPATH(NR,NCOL)=JPSV(NR,NCC)**

** 74 WRITE(6,402)(JPATH(I1,NCOL),I1=1,7)**

** 80 CONTINUE**

** RETURN**

** ENTRY ENDS**

** WRITES LIST OF EQUIPMENT VECTORS FOR OPTIMUM PLANT**
COMPUTES LISTS OF REF DEMANDS, SALES + PSEUDO SERVICE USAGES
ALSO COMPUTES COST SUB-TOTALS

WRITE(6,130)
JCS=JCD=NS=ND=NPS=0

DO 100 KE=1,NEPT
NE=NEOPT(KE)
WRITE(6,132)(EMCH(K,NE),K=1,15)
DO ES 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*

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*

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*

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.0) PSRV(N,2) GO TO 98
97 N=NPS=NS+1
PSRV(N,1)=0.
C
ENTER STREAM + USAGE
98 PSRV(N,1)=PSRV(N,1)+EMCH(10,NE)
PSRV(N,2)=STR
C
CONTINUE
100 CONTINUE

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) REF(D,J,K)=SORT(JJ,K)
105 IF(I.EQ.2) REF(S,J,K)=SORT(JJ+10,K)
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)
**TOTAL PLANT COST SUMMARY**

1. NETWORK (EXCL. REFR. DEMANDS + SALES)
2. PRE-PROCESSING (PRESSURE SPECS)
3. POST-PROCESSING (TEMPERATURE SPECS)
4. REFRIGERATION DEMANDS
5. STREAM SALES

**EQUIPMENT DETAILS**

**JPATH ARRAY (SORTED)**
SUBROUTINE BRBND(LBXX, NREJX, NMIN)

PERFORMS BRANCH + BOUND OPT ON STREAM PROC PATHS (JPATH ARRAY)
ROUTINE ALLOWS UP TO 3 LEVELS OF BRANCHING

COMMON/PATH/JPATH(8,200), NPATH(20), NPTHSH
COMMON/PL0T/NEPT, NOPT(40), NCOST(10)
BLANK COMMON
COMMON NNPR(2), NNJR(2), NNSE(2), SMCH(8,50,2), SMCHB(7,50,2),
1 SMEXH(8,15,100)
COMMON NEMCH, EMCH(15,100)
COMMON COST(500), KTAB(500), NOPL(500)

DIMENSION JA(40), JCNT(40), JSGN(40), JSE(2,20)
DIMENSION JSPB(2), JCRJ(40), JHIST(20)
INTEGER BPP(3), BPR(3,20,3), NBP(3)
DIMENSION LLJ(20), LLN(20), NWP(20), NPA(20), NOC(20)

PARAMETERS -
LBXX - NO OF LEVELS OF BRANCHING
NREJX - MIN NO OF REJECTIONS FOR PROBLEM
NMIN - MIN NO OF PROBLEMS AT ANY LEVEL

JPATH ARRAY - EACH COL REPRESENTS ONE STREAM PROC PATH
1. NO OF EQUIPS IN PATH
2. EQUIP NOS
3. COST OF PATH
4. ACTIVE(0)/INACTIVE(GT 0) FLAG

INDEX DISPLACEMENTS FOR JPATH + NPATH ARRAYS
INDEX DISPLACEMENTS FOR JPATH + NPATH ARRAYS
IDJ(JPR,JCH)=NPTHSH*(JPR-1)+NNPR(1)*(JCH-1)
IDN(JPR,JCH)=JPR+NNPR(1)*(JCH-1)

NPRH=NNPR(1)-NNJR(1)
NPRC=NNPR(2)-NNSE(2)
NPRR=NPRH+NPRC

SET UP COL INDEX VECTORS FOR JPATH, NPATH (1ST PROC COL FOR JPATH)
I=0
DO 8 JCH=1,2
J=1, NNN
DO 8 J=1, NNN
8 CONTINUE

NEGLCT PSEUDO-SERVICE STREAMS
IF(J<=NNJR(JCH)) GO TO 8
L=L+1
LLJ(L)=IDJ(J,JCH)+1
LLN(L)=IDN(J,JCH)
8 CONTINUE

NEGLCT PSEUDO-SERVICE STREAMS
IF(J<=NNJR(JCH)) GO TO 8
L=L+1
LLJ(L)=IDJ(J,JCH)+1
LLN(L)=IDN(J,JCH)
8 CONTINUE

SET SHIFTING CONSTANTS FOR ENCODING PLANT NOS
NOC(NPR)=1
JS=2
DO 10 JS=2,NPR
JC=NPRR+JS-2
10 NOC(JC)=NOC(JC-1)+NPTHSH

JCSTX=NOPX=10000000
NPRX=NPRX2=NPRX3=1
ACT=0
ASSIGN 22 TO IZA
GO TO 260

*** COMPUTE ALL BOUNDING PROBLEMS ***

22 DO 201 IPAS=1,2
LBX=LBXX
IF(IPAS.EQ.1) LBX=1
201 WRITE(6,436) JCSTX, NOPX

** 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 NPR=1,NPRX1
GO TO 200
C
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** SET UP 2ND LEVEL OF BRANCHING **
LB=2
ASSIGN 30 TO IBPR
GO TO 300
30 NPRX2=NBP+1
32 DO 190 NPR2=1,NPRX2
IF(LBX.EQ.1) GO TO 42
NBPI2=NPRI=NPRI2
NBP(3)=0
NPRX=NPRX2
LB=2
ASSIGN 34 TO IBRJ
GO TO 340
34 IF(LBX.EQ.2) GO TO 42
C** SET UP 3RD LEVEL OF BRANCHING **
LB=3
ASSIGN 40 TO IBPR
GO TO 300
40 NPRX3=NBP+1
42 DO 180 NPR3=1,NPRX3
IF(LBX.EQ.1) GO TO 50
NBP(3)=NPRI
NPRX=NPRX3
LB=3
ASSIGN 50 TO IBRJ
GO TO 340
50 IF(IPAS.EQ.2) GO TO 110
C*** COMPUTE CLOSER UPPER BOUND ****
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
ENTER LOWEST COST COMPAT PATH FOR JS
82 ASSIGNED 84 TO I P A R
GO TO 250
84 DO 86 NCOL=NC1,NC2
IF(JPATH8.NCOL).GT.0) GO TO 86
IF(KSE.EQ.J) GO TO 83
C  SEARCH FOR EQUIP NSEQ
NRR=JPATH1.NCOL)+1
DO 95 NR=2,NRR
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 ENTER NEXT OP SIGN INPUT , OTHERWISE MOVE TO NEXT JS
88 KSE=0
IF(JSE1.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)
IDENTIFY JSS 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
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
NWP(JS) = NCOL - NCJ
C
SCAN ALL STREAMS NE JS + ADD IN LOWEST COST ACT PATH
DO 124 LS=1,NPRR
IF(LS .EQ. JS) GO TO 124
NC1 = NCCL = LLJ(LS)
122 NCCL = NCCL + 1
IF(JPATH(8,NCCL) .GT. 0) GO TO 122
NWP(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(I1),I1=1,NREJ)
C
C************************************************************

C*** MAKE UP ALL POSSIBLE PLANTS FROM ACTIVE PATHS ****
C
ISZ = 1
132 DO 134 J=1,NPRR
ISZ = ISZ * NPA(J)
WRITE(6,434) NPA(I1),I1=1,NPRR, ISZ
C
INITIALIZE NPW - TO FIRST ACTIVE PLANT
DO 134 I1=1,NPRR
NCOL = NCJ = LLJ(I1)
133 NCOL = NCOL + 1
IF(JPATH(8,NCOL) .GT. 0) GO TO 133
134 NPW(I1) = NCOL - NCJ
NN = NNSEP(1) + 1
NPT1 = NPATH(NN)
NPL = 0
GO TO 142
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
C IF(NPW(1) .GT. NPT1) GO TO 150
C IF NPATH(NCN) EXCEEDED , RE-SET TO 1
C IF(NPW(JS) .GT. NPS) NPW(JS)=IEX=1
C NPW=NCJ+NPW(JS)
C IF(JPATH(8+NCOL) .GT. 0) GO TO 138
140  IF(IEX.EQ.0) GO TO 142
C COMPUTE PLANT NO + COST + ENTER IN SORT VECTORS - IF COST LT BOUND
142  ASSIGN 144 TO INOP
GO TO 210
144  IF(I CST .GE. J CSTX) GO TO 136
NPL=NPL+1
CST(NPL)=ICST
NOPL(NPL)=NOPL
GO TO 136
C WRITE(6,418) NPL
C***************
C** SORT **
CALL TGSORT(COST,KTAG,NPL,-1)
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  JCSTX=ICST
NOPX=NOPL
WRITE(6,412) JCSTX, NOPX, L
GO TO 176
160  CONTINUE
C WRITE(6,422)
GO TO 176
175  WRITE(6,425) NCOL
C 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*--------------------------------------------------------------------------
**COMPUTE + SORT EQUIP NO VECTOR FOR OPT PLANT (INCL PRE-PROC EQUIP)**

```plaintext
NCOST(1)=JCSTX
NCP=NOPX
ASSIGN 202 TO INPW
GO TO 220
202
NEP=0
DO 208 JS=1,NPPR
NCOL=LLJ(JS)
DO 208 M=1,2
IF(M.EQ.2) NCOL=NCOL+NPW(JS)
NRR=JPATH(1,NCOL)+1
IF(NRR.EQ.1) GO TO 208
DO 206 NR=2,NRR
NEQ=JPATH(NR,NCOL)
IF(NEQ.EQ.0) GO TO 205
HAS NEQ BEEN ENTERED
DO 204 N=1,NEPT
204 IF(NEQ.EQ.1) IFIX(COST(N)) GO TO 206
205 NEPT=NEPT+1
COST(NEPT)=NEQ
206 CONTINUE
208 CONTINUE

SORT INTO INCR ORDER
CALL TGSORT(COST,KTAG,NEPT,-1)
DO 209 J=1,NEPT
209 JPATH(J)=COST(J)
WRITE(6,420)NOPX,JCSTX,(NEOPT(I)),I=1,NEPT
RETURN
```

**ROUTINE TO COMPUTE NOP + ICST (PLANT NO + COST) FROM NPW VECTOR**

```plaintext
210 NOP=ICST=0
DO 212 KK=1,NPPR
KCOL=NPW(KK)+LLJ(KK)
NOP=NOP+NPW(KK)*NOC(KK)
212 ICST=ICST+JPATH(7,KCOL)
GO TO INOP,(98,125,144)
```

**ROUTINE TO COMPUTE NPW VECTOR FROM NOP**

```plaintext
220 NOPP=NOP
DO 222 KK=1,NPPR
NPW(KK)=(NOPP/NOC(KK))
222 NOPP=NOPP-(NPW(KK)*NOC(KK))
GO TO INPW,(154,202)
```

**ROUTINE TO ENTER COL NCOL (STREAM JS) INTO PLANT + CHECK COMPAT**

** IF COMPATIBLE ICOMP IS SET TO 1 **
** INPUTS ARE ENTERED IN JSA VECTOR + ENTRIES COUNTED IN JCNT VECTOR **
** COUNT Ix=1 FOR PROC/PROC MATCH ; 2 FOR PROC/SERV MATCH V/R OR SALE **
** SIGN OF JS FOR ENTRY IS STORED IN JSGN VECTOR **
** TEST ON NPAS=1 , ENTER ON NPAS=2 **

```plaintext
230 ICOMP=0
IF(NSA.EQ.0) CALL ZEROI(JSA,160)
NHC=2*(NPRH=JS)+1
NRR=JPATH(1,NCOL)+1
DO 240 NPAS=1,KPAS
DO 242 NR=2,NRR
NEQ=JPATH(NR,NCOL)
IX=1
DO 242 I=3,4
NS=EMCH(I,NEQ)
240 IF(NS.EQ.0) OR NS GT 200) IX=2
DO 239 NI=3,4
NS=EMCH(NI,NEQ)
242 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* 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
   - 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* 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=NSEH+1
   JSE(1,NSE)=NEQ
   JSE(2,NSE)=NS
   239 CONTINUE
   240 CONTINUE
   ICOMP=1
   246 GO TO JSENT,(85,156)

C**** ROUTINUE TO SET UP PARAMS FOR PATH SCAN FOR STREAM JS **
250 NCJ=1.LJ(JS)
   NCN= LLN(JS)
   NPS= NPATH(NCN)
   NC1=NCJ+1
   NC2=NCJ+NPS
   GO TO IPAR,(184,121,137,264,302,342)

C**** ROUTINUE TO SUPPLY JS FOR STREAM JSS **
   ALSO SUPPLIES SEC STREAM NO + ACTIVE/INACTIVE FLAG
255 JCH=1-1SIGN(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+NPB-NNSER(2)
   GO TO 1JS,(94,305,308,366)

C**** ROUTINUE TO ZERO ACT FLAGS PRES GE IACT + RE-SET NPA VECTOR **
260 DO 266 KK=1,NPBR
   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(KCOL).GE.IACT) JPATH(KCOL)=0
   266 IF(JPATH(KCOL).GE.0) NPA(KK)=NPA(KK)-1
   GO TO 1ZA,(22,180,190,200,316)

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
DISCARD PROBLEMS WHICH PRODUCE LT NREJX REJECTIONS

300 NBPR=JPAS=0
LR1=LR2=LB+1
IF(LB.EQ.3) LR1=LR1-1
301 DO 322 MS=1,NPRR
JS=M5
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 BPPC1=NEQ=JPATH(LBB,MCOL)
IF(NEQ.EQ.0) GO TO 320
IF(IVR.EQ.1) GO TO 307
CC IF EITHER INLET STREAM INACTIVE . SKIP COL
C (INACTIVE INLET - 2ND STAGE V/R COMPRESSOR OR A/C INLET)
C SKIP ALSO IF SEC STREAM NO FOR EITHER IS INVALID FOR LEVEL LB
DO 306 I=3,4
BPPC1=JS=EMCH(I,NEQ)
C REJECT IF NOT PROC/PROC MATCH OR V/R COMPRESS (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 IF OUTLET STREAM INACTIVE , SKIP TO NEXT ROW
C (INACTIVE OUTLET - V/R COMPRESS OR 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) BPPC3=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(LLL)
312 IF(NEQ.EQ.BPR(1,KPR,LL)) GO TO 318
ENTER MATCH IN BPR - IF NOT ALREADY ENTERED
313 DO 314 NN=1,NBP
314 IF(BPR(1,NN,LB).EQ.NEQ) GO TO 318
NPR=NBP=NBP+1
DO 317 I=1,3
317 BPR(I,NPR,LB)=RPP(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
CONTINUE
FOR N BPR LT NMIN , RE-SCAN FOR 1. PROBLEMS
IF(NBPR.GE.NMIN. OR. JPAS.EQ.1) GO TO 324
JPAS=1
LRI=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 1BP1,(24,30,40)

****
** SCAN PATHS FOR INFEAS DUE TO MULT STREAM USE ****
** (STREAMS IN BPR BUT NOT IN KEQ MATCH) - SET INACT FLAG
** MUST ALSO CHECK THAT BPR IS PRESENT

340 WRITE(6,408)(NBP(II),II=1,LBX)
NRJ=0
IF(NPR.EQ.NPRX) GO TO 372
SET UP BPR + IDENTIFY JPATH SECTION(S) CONTAINING BPR
376 KEQ=BPR(1,NPR,LB)
DO 370 J=1,2
JS(S)=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
DO 362 JS=1,NPRR
ASSIGN 342 TO IPAR
GO TO 250
342 DO 360 NCOL=NC1,NC2
IF(JPATH(0,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 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
C 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(* 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,*)
436 FORMAT(* BEST BOUND FROM PASS 1 *,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.
DO 20 ALL(REFRIGERANT) CIRCUITS

SET CIRCUIT PARAMETERS

15

TRANSFER PURCHASED STREAM RESIDUALS TO NEXT CIRCUIT

20

COMPUTE COST FIGURES FOR ALL LEVELS

WRITE OUTPUT DATA

RETURN

100

SET UP UNIT FLOW SATURATED LIQUID STREAM

COMPUTE PRESSURE LEVELS FROM TEMPERATURE LEVELS

105

START TOTAL (CIRCUIT) FLOW ITERATION

COMPUTE LIQUID COOLING BY PURCHASED STREAMS

Figure II.3 RUNIT Algorithm (Continued on page 166).
Figure II.3 HUNIT Algorithm (Continued).
PROGRAM MAINR (INPUT=1001, OUTPUT=1001, TAPE5=INPUT, TAPE6=OUTPUT)

*** COMMON DECK ***

COMMON/CONT/L/NEX, NIN, NOUT, NOCOMP
COMMON/PROP/COMPNT(8), APC(8), ATO(8), AVC(8), AMS(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/PARM/AMORT, HRS, TWAT, DTW, CWAT, TS, HVS, CS, CKWH, APPP, APRR,
1 ARR, TRR, DIF(2)
COMMON/SPLINE/NH, NC, X(10), Y(10), P(10)
COMMON/REFD/LRF(3), NLL, RLEV(10, 2), TMSERV

COMMON NSMRB(3), SMRB(7, 20, 3), SMRX(8, 4, 3)

***

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 ARR, TRR, DIF
NAMELIST/COMP/NOCOMP, COMPNT

READ COMPONENT INFORMATION
READ(5, COMP)
READ COMPONENT PHYSICAL CONSTANTS
DO 10 I = 1, NOCOMP
10 READ(5, 11)(PROP(I, K), K = 1, 15)

READ TITLE
READ(5, 100) TITLE
IF (EOF(5), 1, 2)
1 CALL EXIT
2 WRITE(6, 101) TITLE
READ SYSTEM PARAMETERS
READ(5, PARLST)

READ IN PRESENT REF LEVELS + COSTS
READ(5, 700) NLL
DO 704 I = 1, NLL
704 READ(5, 702)(X(I), Y(I))

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)

READ IN NEW REF LEVELS
READ(5, 700) LRF, NLL
DO 720 I = 1, NLL
720 READ(5, 702)(RLEV(L, II), II = 1, 2)

READ IN SOLID STREAM INFORMATION + PLACE IN REFS ARRAYS
CALL ZER0(1, NSMRB, 3)
READ(5, 700) NS
DO 730 N = 1, NS
READ(5, 700) IR
NN = NSMRB(IR) = NSMRB(IR) + 1
READ PROPS
READ(5, 702)(SMRB(I, NN, IR), II = 1, 7)
READ(5, 702)(SMRX(I, NN, IR), II = 1, 7)
READ(5, 702) TMSERV

CALL RUNIT
11 FORMAT(3/(5E14.5))
100 FORMAT(8A10)
101 FORMAT(8A10/)
700 FORMAT(4F5.1)
702 FORMAT(7F10.0)
END
SUBROUTINE RUNIT
COMPUTES THREE-TIER CASCADE REFRIGERATION SYSTEM
REFRIGERANTS - METHANE, ETHYLENE, PROPANE
UP TO 3 LEVELS OF EACH REFRIGERANT (PURE REFRS ASSUMED)

SUBSCRIPT IR, 1 METHANE, 2 ETHYLENE, 3 PROPANE
RLEV - REFR DEMAND VECTORS 1. T, 2. Q
NLEV - NO OF LEVELS
EMR - EQUIP MATRIX
SMRB - STREAM MATRIX (B SECTION)

***** 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/DPOUT(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,IRS,TWAT,DTW,CWAT,T,HVS,CS,CKWH,APPP,APRR,
1 ARR,TRR,DTF(2)
COMMON/NSMRB(3),SMRB(7,20,3),SMRB(8,4,3)
COMMON/NEMR(3),EMR(15,15,3)

DIMENSION TEX(2),DUM(8),DHV(3),SUMQ(3),COEF(4,3)
DIMENSION TLEV(10),QLEV(10),PLEV(10),FLEC(2),QRC(2)
DIMENSION NCM(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 NCM(2,3,6)/
DATA TRFC/305.,425.,545./

P VS T FUNCTION FOR PURE REFERS
P(IR,T)=COEF(1,IR)+COEF(2,IR)+COEF(3,IR)+COEF(4,IR))

ZERO MATRICES + MATRIX COUNTERS
CALL ZEROI(NMR+3)
CALL ZEROI(EMR,675)
CALL ZEROI(FLEC,10)
CALL ZEROI(COST,16)
CALL ZEROI(SUMQ,3)

SET PURCHASED STREAM COUNTER
PURCHASED STREAMS ARE ALREADY STORED IN REFERENCE ARRAYS
DO 10 IR=1,3
10 NSP(IR)=NSMRB(IR)

*** SET UP + COMPUTE SECTIONS (METHANE - PROPANE)
NL2=0
DO 20 IR=1,3
20 IF(LRF(IR),EQ,0) GO TO 20
LLV(1,IR)=NL1=NL2+1
LLV(2,IR)=NL2=NL1+LRF(IR)-1
ICLC=0
NC=NCM(1,IR)
TCOND=TRFC(1,IR)+APR
GO TO 100
15 IF(IR,EQ,3) GO TO 20

STORE CONDENSING LOAD DATA (METHANE, ETHYLENE)
FLOC(IR)=FLO
QRC(IR)=QCOND
ADD COND LOAD TO LOWEST LEVEL LOAD FOR NEXT REFR
QLEV(NL2+1)=QLEV(NL2+1)+QCOND

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,NSP
C LOAD SOLD STREAM RESIDUALS INTO SIN(1), THEN INTO SMRB
NN=NRSS(JS)
CALL MVFSM(1,NN,IR+2,JS)
ASSIGN 16 TO LOADS
GO TO 400
16 NSP(IR)=NSMRB(IR)
20 CONTINUE
C*** COMPUTE TOTAL SYSTEM COSTS - INCL CPS*CPE COSTS
25 CTOT=0.
28 CTOT=CTOT+COST(L)
DO 30 IR=1,3
30 CTOT=CTOT+CPS(IR)+CPE(IR)
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(1,13)) IR=3
COST(L)=(COST(L)/QLEV(L)+(CPS(IR)+CPE(IR))/SUMQ(IR))/HRS
32 CONTINUE
C ADD CONDENSATION COSTS
DO 36 IR=1,2
IF(LRF(IR).EQ.0.) GO TO 36
LEV=LLV(1,IR+1)
QCOST=ORC(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=NSMRB(IR)
KSM=NSMRB(IR)
WRITE(6,963)
DO 970 K=1,KSM
WRITE(6,972)K,SMRB(KK,K,IR),KK=1,7
WRITE(6,955)
DO 960 K=1,KEQ
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**** REFRACTORY SECTION COMPUTING ROUTINE ****
C SET SIN(1) AS UNIT FLOW STREAM (SAT LIQ) AT COND PRES
100 CALL ZERO(XIN,NCOMP)
TMIN(1)=XIN(NC,1)=1.
C COMPUTE CONDENSATION PRESSURE
TIN(1)=BPIN(1)=TCOND
VFIN(1)=0.
PIN(1)=P(IR,TCOND)
WRITE(6,817)TIN(1),PIN(1)
CALL PUBL(1,PCOND,DUM)
PINF(1)=PLEV(NLL+1)=PCOND
CALL EMTH(1,HIN(1),DUM)
C LOAD INTO SMRB NLIQ
ASSIGN 103 TO LOADS
GO TO 400
103  NLIQ=NSM

C  COMPUTE PRESSURE LEVELS + SUM QS
DO 104 L=NL1,NL2
   TIN(1)=TELV(L)
   PIN(1)=P(IR*,TELV(L))
   WRITE(6,817)TIN(1),PIN(1)
   CALL PUBL(1,PLEV(L),DUM)
104   ESTIMATE REF CIRCULATION
FLO=SUMQ(IR)+QLEV(L)
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** COMPUTE LIQ COOLING BY PURCHASED STREAMS
C
   NSL=NLIQ
   XFLO=FLOW
   DO 118 JP=1,NNSP
      ORDERING OF PURCHASED STREAMS
      NP=JP
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
      CALL MVFSM(2,NP,IR2,NP)
      C EXCHANGE , LOAD STREAMS + EQUIP
      JIN1=NS1
      JIN2=NP
      CALL HXER(1,4,TEX,Q)
      CALL ASSIGN(122 TO LOADS)
      GO TO 400
C 112   NRSS(NP)=NSM
C 113   COMPUTE VALUE OF COOLING USED
      DT=DTF(2)*APPR
      CALL MVFSM(1,NP,IR2,NP)
      CALL SVALUE(2,TOUT(2),DT,VAL)
      CPS(IR)=CPS(IR)+VAL
      NSL=NSM-1
      CPE(IR)=CLEV
      GO TO 122
C 120   NSL=NSM
C 122   SAVE COUNTERS
      NSMSV=NSM
      NEMSV=NEM
      TSL=SMRB(3,NSL,IR)
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
      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
      CALL ADBF(PLEV(L))
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
INV=-1
XFLO=FXN
ASSIGN 126 TO IFLO
GO TO 440
C

C LOAD STREAMS + EQUIP FOR FLASH
JIN2=0
IF(IR.EQ.3) GO TO 128
C IS CROSS EX POSSIBLE
DTX=(TSL-TOUT(1)-(1.*APRR)
IF(DTX.GT.0.) GO TO 130

C* PROPANE
JIN1=NSL
ASSIGN 142 TO LOADS
GO TO 400
C*
126 IR=1,2 - REQUIRES ITERATION OF FLASH/CROSS EX LOOP
C RESET COUNTERS BEFORE LOADING
130 NSM=NSMR6(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 MVSOSt(2,1,0,0)
CALL MVFSM(1,NSL,IR2,NLIO1)
INV=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 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* 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 SET NSM TO NVP(NL1) + LOAD INTO SIN(1)
152 NSM=NVP(NL1)
CALL MVFSM(1,NSM,IR2,NLIQ)
FLO=0.
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)
Q0=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)
JIN=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=TN(1)-(TWAT+DTW+APPPL)
IF(DTA.LT.0.) GO TO 170
C**
AFTERCOOL ; LOAD STREAMS + EQUIP
JIN1=NSM
JIN2=202
TEX(1)=AMAX1((TWAT+APPPL),(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*
CONDENSE ; LOAD EQUIP ONLY
172 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 QCND=ABS(Q)
C
PROPORTION + SUM COSTS
180 DO 182 LL=NL1,L
182 COST(LL)=COST(LL)+CLEV*(FLEV(LL)/FLO)
WRITE(6,990)CLEV,(COST(LL)),LL=NL1,L
200 CONTINUE
GO TO 15
C
*** OUTPUT STREAM LOADING ROUTINE ***
C 400 IR2=IR+2
   IF (ICALC.EQ.1) GO TO 402
   JO=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
   STORE MOLE FRACTIONS IN SMRX
   DO 412 I=1,NOCOMP
412 SMRX(I,NSM,IR)=XIN(I,1)/TMIN(I)
   GO TO 430
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)
   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*** 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 -*, 15, 5X, 2F8.1, F10.0, F8.1//)
517 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, 5X), F12.0, 5F9.1, 3F9.0)
963 FORMAT(* STREAM MATRIX-*/)
972 FORMAT(15, 3F8.1, 5X, 3F12.8)
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)
STREAM MOVING UTILITY ROUTINE ...(RUNIT VERSION)
IWV - ELEMENT NUMBER IN SIN OR SOUT WORKING ARRAY
   + SIN
   - SOUT
ISM - VECTOR NUMBER IN SM--
III - VECTORS MOVE TO OR FROM SMPB
1-2 MOVE TO OR FROM SMCHB - (1-2)
3-5 MOVE TO OR FROM SMRB - (1-3)
NX - VECTOR NUMBER CONTAINING MOLE FRACTIONS (III GT 0)
3 ENTRIES -
1 MVSOI MOVES SOUT VECTOR ISM TO SIN VECTOR IWV (III=0)
2 MVFSM MOVES FROM SM-- TO SIN OR SOUT
3 MVTSM MOVES TO SM-- FROM SIN OR SOUT

*** COMMON DECK ***
COMMON/CONT/L/NE,NIN,NOUT,NOCOMP
COMMON/SIN/DPIN(4),DPIN(4),PIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
1 XIN(8,4)
COMMON/SOUT/DPOUT(4),DPOUT(4),DOUT(4),DOUT(4),HOUT(4),VOUT(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** ENTRY MVSOI
IENT=1
GO TO 1
C** ENTRY MVFSM
IENT=2
GO TO 1
C** ENTRY MVTSM
IENT=3
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
5 GO TO (100,200,300)IENT
C** MVSOI
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** MVFSM
200 DO 10 I=1,7
10 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** 300 MVTSM
300 DO 30 I=1,7
   IF(IWV.GT.0) AA=SIDUM(IWV,1)
   IF(IWV.LT.0) AA=SODUM(-IWV,1)
   GO TO (21,22,23)IYPE  
21 SMPB(1,ISM)=AA
   GO TO 30
22 SMCHB(I,ISM,KKK)=AA
   GO TO 30
23 SMRB(I,ISM,KKK)=AA
30 CONTINUE
   IF(IYPE.GT.1) RETURN
   DO 40 I=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 (ADB) 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=PUNCH

PHYSICAL PROPERTIES DATA LIBRARY ROUTINE
PUNCHES OUT DATA FOR SELECTED COMPONENTS

PURE COMPONENT ID NUMBERS...

1. HYDROGEN
2. METHANE
3. ETHANE
4. PROPANE
5. I-BUTANE
6. N-BUTANE
7. I-PENTANE
8. N-PENTANE
9. N-PENTANE
10. N-HEXANE
11. N-HEPTANE
12. N-OCTANE
13. N-DECANE
14. N-UNDECANE
15. N-DODECANE
16. N-TRIDECANE
17. N-TETRADECANE
18. N-PENTACANE
19. N-HETAPSEDADECANE
20. N-HETAPSEDADECANE
21. PROPYLENE
22. ETHYLENE
23. 1-BUTENE
24. 1-BUTENE
25. cis-2-PENTENE
26. trans-2-PENTENE
27. 1-PENTENE
28. cis-2-PENTENE
29. 1-PENTENE
30. cis-2-PENTENE
31. trans-2-PENTENE
32. 1-PENTENE
33. 1-PENTENE
34. 2-METHYL-2-BUTENE
35. 1-HEXENE
36. METHYLCYCLOPENTANE
37. CYCLOHEXANE
38. METHYLCYCLOHXANE
39. ETHYLBENZENE
40. BENZENE
41. TOLUENE
42. CYCLOPENTANE
43. M-XYLENE
44. P-XYLENE
45. ETHYL B走去HENE
46. NITROGEN
47. OXYGEN
48. CARBON MONOXIDE
49. CARBON DIOXIDE
50. HYDROGEN SULFIDE
51. SULFUR DIOXIDE
52. 2-METHYL-C5
53. 3-METHYL-C5
54. 2,2-DI-C1-C4
55. 2,3-DI-C1-C4
56. 3-METHYL-C5
57. PROPAINE
58. 1,2-BUTADIENE
59. 2-METHYL-1-BUTENE
60. C2-CYCLO-C6
61. ISOPRENE
62. WATER

STANDARD COMPONENT NAMES

INTEGER SCNAM(248)
DATA (SCNAME(I), I=1,157)/

COMMON/PROP/APC(10), ATC(10), AVC(10), AMW(10), AOMEG(10), ADEL(10),
IAVW(10), APH(10), BET(10), GAM(10), DTA(10),
BASEA(10), BASEB(10), ZCD(10), ALD(10)/

CHAO-SEADER MODIFIED ACENTRIC FACTORS - DIMENSIONLESS
REAL OMEGA(62)
DATA OMEGA(C), OMEGA(C)

1. HYDROGEN
2. METHANE
3. ETHANE
4. PROPANE
5. I-BUTANE
6. N-BUTANE
7. I-PENTANE
8. N-PENTANE
9. N-PENTANE
10. N-HEXANE
11. N-HEPTANE
12. N-OCTANE
13. N-DECANE
14. N-UNDECANE
15. N-DODECANE
16. N-TRIDECANE
17. N-TETRADECANE
18. N-PENTACANE
19. N-HETAPSEDADECANE
20. N-HETAPSEDADECANE
21. PROPYLENE
22. ETHYLENE
23. 1-BUTENE
24. 1-BUTENE
25. cis-2-PENTENE
26. trans-2-PENTENE
27. 1-PENTENE
28. cis-2-PENTENE
29. 1-PENTENE
30. cis-2-PENTENE
31. trans-2-PENTENE
32. 1-PENTENE
33. 1-PENTENE
34. 2-METHYL-2-BUTENE
35. 1-HEXENE
36. METHYLCYCLOPENTANE
37. CYCLOHEXANE
38. METHYLCYCLOHXANE
39. ETHYLBENZENE
40. BENZENE
41. TOLUENE
42. CYCLOPENTANE
43. M-XYLENE
44. P-XYLENE
45. ETHYL B走去HENE
46. NITROGEN
47. OXYGEN
48. CARBON MONOXIDE
49. CARBON DIOXIDE
50. HYDROGEN SULFIDE
51. SULFUR DIOXIDE
52. 2-METHYL-C5
53. 3-METHYL-C5
54. 2,2-DI-C1-C4
55. 2,3-DI-C1-C4
56. 3-METHYL-C5
57. PROPAINE
58. 1,2-BUTADIENE
59. 2-METHYL-1-BUTENE
60. C2-CYCLO-C6
61. ISOPRENE
62. WATER

STANDARD COMPONENT NAMES

INTEGER SCNAM(248)
DATA (SCNAME(I), I=1,157)/

COMMON/PROP/APC(10), ATC(10), AVC(10), AMW(10), AOMEG(10), ADEL(10),
IAVW(10), APH(10), BET(10), GAM(10), DTA(10),
BASEA(10), BASEB(10), ZCD(10), ALD(10)/

CHAO-SEADER MODIFIED ACENTRIC FACTORS - DIMENSIONLESS
REAL OMEGA(62)
DATA OMEGA(C), OMEGA(C)

1. HYDROGEN
2. METHANE
3. ETHANE
4. PROPANE
5. I-BUTANE
6. N-BUTANE
7. I-PENTANE
8. N-PENTANE
9. N-PENTANE
10. N-HEXANE
11. N-HEPTANE
12. N-OCTANE
13. N-DECANE
14. N-UNDECANE
15. N-DODECANE
16. N-TRIDECANE
17. N-TETRADECANE
18. N-PENTACANE
19. N-HETAPSEDADECANE
20. N-HETAPSEDADECANE
21. PROPYLENE
22. ETHYLENE
23. 1-BUTENE
24. 1-BUTENE
25. cis-2-PENTENE
26. trans-2-PENTENE
27. 1-PENTENE
28. cis-2-PENTENE
29. 1-PENTENE
30. cis-2-PENTENE
31. trans-2-PENTENE
32. 1-PENTENE
33. 1-PENTENE
34. 2-METHYL-2-BUTENE
35. 1-HEXENE
36. METHYLCYCLOPENTANE
37. CYCLOHEXANE
38. METHYLCYCLOHXANE
39. ETHYLBENZENE
40. BENZENE
41. TOLUENE
42. CYCLOPENTANE
43. M-XYLENE
44. P-XYLENE
45. ETHYL B走去HENE
46. NITROGEN
47. OXYGEN
48. CARBON MONOXIDE
49. CARBON DIOXIDE
50. HYDROGEN SULFIDE
51. SULFUR DIOXIDE
52. 2-METHYL-C5
53. 3-METHYL-C5
54. 2,2-DI-C1-C4
55. 2,3-DI-C1-C4
56. 3-METHYL-C5
57. PROPAINE
58. 1,2-BUTADIENE
59. 2-METHYL-1-BUTENE
60. C2-CYCLO-C6
61. ISOPRENE
62. WATER

STANDARD COMPONENT NAMES

INTEGER SCNAM(248)
DATA (SCNAME(I), I=1,157)/

COMMON/PROP/APC(10), ATC(10), AVC(10), AMW(10), AOMEG(10), ADEL(10),
IAVW(10), APH(10), BET(10), GAM(10), DTA(10),
BASEA(10), BASEB(10), ZCD(10), ALD(10)/

CHAO-SEADER MODIFIED ACENTRIC FACTORS - DIMENSIONLESS
REAL OMEGA(62)
DATA OMEGA(C), OMEGA(C)
CHAO-SEADER MODIFIED HILDEBRAND SOLUBILITY PARAMETER

(CAL./ML.)* 1/2

CHAO-SEADER CHARACTERISTIC MOLAR VOLUMES - ML./ G-MOLE

CHAO-SEADER MODIFIED HILDEBRAND SOLUBILITY PARAMETER

(VOLUME AT 25 DEG.C. / ML./ G-MOLE)

CRITICAL TEMPERATURES, DEG. K.

CRITICAL PRESSURES, ATM.

CRITICAL VOLUMES, CC./G-MOLE

MOLECULAR WEIGHTS

DENSITIES AT 15 DEG. C. / G./ML.

*** COEFFICIENTS OF ZERO PRESSURE HEAT CONTENT.  *****

REAL APHA(62)
DATA APHA/  6.9523, 3.3812, 2.2472, 2.4103, 3.3324, .4534, .8165, 9.1043, 4.37
17.4779, .0551, 10.6262, 12.1983, 13.7705, 15.3421, 16.9148, 4.8620, .9642, 21
263.23, 202.24, 774.94, 753.8, 24.1, 778.2, 34.1, 65.1, 12.9, 1.7883, 3.35
311.49, 4.953, 270.13, 2063, 12.9577, 12.1411, 15.9355, 15.075, 8.655
4-8.213, -3.788, -6.533, -5.334, -8.398, 9.903, 6.085, 6.7265, 3.1677, 0.767, 5
5157.1, 361.2, 621.5, 593.1, 298.2, 344.3, 0.1592, 2.8487, -12.282, -15.559
6.6687, 7.707/
REAL BETTA(62)
DATA BETTA/ -.04576E-2, 18.004E-3, 38.201E-3, 57.195E-3, 75.214E-3,
28E-3, 152.248E-3, 168.198E-3, 184.148E-3, 200.098E-3, 216.048E-3, 231.014
4078E-2, 7.22E-2, 7.702E-2, 8.35E-2, 1.101E-3, 105.623E-3, 99.696E-3, 10
7.935E-2, 10.375E-2, 3631E-2, .04001E-2, 1.4285E-2, 3.128E-2, 1.1384E-2,
81.5712E-2, 12.3504E-2, 21.13E-2, 12.6929E-2, 14.4802E-2, 4.503E-2,
REAL GAMA(62)
DATA GAMA/ .09563E-5, -4.3E-7, -1.1049E-7, -1.7533E-7, -2.3734E-7,
-2425.93E-7, -4.7662E-7, -5.2731E-7, -5.783E-7, -6.26869E-7, -6.76938E-7,
-30.02E-7, -4.7687E-7, -6.6276E-7, -8.3145E-7, -8.8821E-7, -1.9935E-7,
-2.91E-8, -5.11E-8, -4.407E-8, -3.490E-8, -3.981E-8, -5.582E-8, -5.542E-8,
-7.28E-8, -8.0344E-7, -9.425E-7, -1.574E-7, -1.04E-7, -9.93E-8, -7.54E-8,
-9.784E-8, -1.0755E-7, -1.1930E-5, -1.1970E-5, -1.828E-5, -8.365E-6, -2.11E-7,
-7.8684E-5, -9.542.2, 5.56E-5, -9.418E-5, -1.2304E-5, -1.2188E-5, -5.536E-5,
-2.521E-6/
REAL DELTA(62)
DATA DELTA/ -.2079E-9, .20E-9, .22E-9, -9.5, 8.8E-9, 12.07E-9, 7.89E-9.6,
107E-9.8, 0.2E-9, 14.24E-9, 11.858E-9, 12.911E-9, 10.948E-9, 9.124.91E-9,
2041E-9, 9.65E-9, 14.44E-9, 16.41E-9, 20.03E-9, 11.27E-9, 24.03E-9, 18.54E
3.9, 18.0E-9, 18.4.8E-9, 2.0, 0.4E-9, 1.70E-9, 23.09E-9, -6.48E1E-9, 3533E-9,
4.5.307E-9, 1.78E-9, 8.786E-9, 2.057E-9, .94.0, 17.18E-9, 5.678E-9, 7.043
5E-9, 22.835E-9, 26.997E-9, 1.2629E-8, 8.587E-10/

C
READ COMPONENT NUMBER
800 READ(5,712)
I=EOF, 5)72O, 721
720 CALL EXIT
721 I=4*I-1)+1
I2=I1+3
VC(I)=VC(I)*.45359/28.32
PUNCH DATA
WRITE(7,712), (SCNAME(JJ), JJ=11,2)
PP=14.696*PC(I)
TT=1.8*TC(I)
BASEB(I)=0.867*TT/PP
ZCD(I)=PP*VC(I)/(10.73*TT)
BASEA(I)=SQRV(0.4278*TT*2.5/PP)
WRITE(7,714), PC, TC, VC(I), MW(I), OMEGA(I), DEL(I), VW(I), APHA(I), BETTA(I),
GAMMA(I), DELTA(I), BASEA(I), BASEB(I), ZCD(I), DENS(I)
GO TO 800
C
712 FORMAT(15,4A4)
714 FORMAT(5E14.5)
END
SUBROUTINE KHZT(ARG,ANS,List)

THIS IS A COMPREHENSIVE THERMO. DATA SUBROUTINE WITH 8 ENTRY POINTS

1. ENTRY DENS (ARG,ANS,AMW)
2. ENTRY ENTH (ARG,ANS,DUM)
3. ENTRY KVAL (ARG,ANS,LIST)
4. ENTRY TSBH (ARG,ANS,DUM)
5. ENTRY BUBMP (ARG,ANS,LIST)
6. ENTRY DEWP (ARG,ANS,LIST)
7. ENTRY PUBL (ARG,ANS,LIST)
8. ENTRY PEDW (ARG,ANS,LIST)

***** COMMON DECK
COMMON/CONTINE,NBIN,NOUT,NOCOMP
COMMON/SIN,DPIN(4),TPIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
1 XIN(8,4)
COMMON/SOUT/DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
1 TMOUT(8,4)
COMMON/PROP/COMPNT(8),APC(8),ATC(8),AVC(8),AMW(8),AOME(S),
1 DEL(8),AVW(8),AFPH(8),BET(8),GAM(8),DATA(8),BASEA(8),BASEB(8),
2 ZCD(8),ALD(8)

***** INTEGER ARG,COUNT,COUNT,DOUNT,COMPNT,VPFRAC
REAL X18J,LIST81,LNP181,LPN181,KV(8),NEWX(8),AV25(8)
REAL LHC,MPOLY,MAWDY
LOGICAL FFLAG,FLAG,AKFLAG
EQUIVALENCE ITRE,TEMUR
DATA TTLOW/240.1

INTERNAL FUNCTIONS

***** DELHV + DELHL *****
FUNCTION FOR TEMUR STARTING VALUE FOR BUBBLE + DEW POINT ITERATION
REGRESSED OVER RANGE AAMW=11.55 , PRSSUR=10.565 (HYDROCARBONS CO-C
CHAO-SEADER COEFFICIENTS FOR LIQUID FUGACITY
AS MODIFIED BY GRAYSON AND STRUED.
CHAO-SEADER COEFFS RETAINED FOR H2,CH4
REAL COEFFT/3.101
DATA COEFFT/196718.243840,2.05135,102972,2.24550,2.10899,
1 0.054009,0.34084,0.005288,0.00212,0.19396,0.00223,
2 0.02282,0043585,0.10486,0.08852,0.036913,0.00872,2*0,
3 -0.00353,2*0,0.002037/.

REAL FRI(5,3)
DATA FRI/-.0817,3274,.-5014,3870,.-1342,0.093,.-3445,4.042,
1 -1.0168,0.05473,0.0894,.-4344,7915,.-7654,3367/.
REAL FRJ(5,3)
DATA FRJ/-.0230,.-0124,1625,.-2135,.-08643,.-022,.-003363,
1 -0.0796,.-08546,.-0217,0.0674,.-06109,.-06261,.-2378,.-1655/
REAL FRK(4,3)
DATA FRK/0.05626,.-3518,6194,.-3809,0.01937,.-03055,0.06310,0.0,
1 -0.01393,.-003459,.-16117/.
REAL FRL(5,3)
DATA FRL/-.21.055,174,-33.637,-28.109,26.277,-16.0,30.699,
1 19.645,-81.305,47.031,-6.55,7.8027,15.344,-37.042,2,1697.

**** PRESET VALUES ****
DATA LNP181,LPN181,LNP181/2410.

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=PRESSUR/(10.73*TEMn*m*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(80H*** DENSITY CANNOT BE CALCULATED BECAUSE VAPOR FRACTION
1 IS IMPROPERLY SPECIFIED)
   ANS=0.5
   RETURN

ENTRY ENTH
   ASSIGN 30 TO LOC
   GO TO 1000
31 ANS=0.
   RETURN
30 IF (TEMm+1 LT.1.) GO TO 31
   LOS=1
   GO TO 14000
33 ANS=GETH
   RETURN

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(71H*** K-VALUES CANNOT BE CALCULATED, VAPOR FRACTION IMPRO
PERLY SPECIFIED)
47 DO 48 I=NFC,NLC
48 LIST(1)=KV(I)
   RETURN

ENTRY TSUBH
   ICAL=1
   ASSIGN 50 TO LOC
   GO TO 1000
50 IF (TEMm+EQ.0.) TEMm=500.
   FRDv=0.10
   DO 56 COUNT=1,10
      LOS=2
      GO TO 14000
52 HTRY=GETH
   SUMm=HCON-HTRY
   IF (ABS(SUMm/HCON) .GT. 1.E-3 ) GO TO 55
   ANS=TEMm
   RETURN
55 ASSIGN 56 TO LOC
   GO TO 2000
56 CONTINUE
   WRITE(6,57)
57 FORMAT(54H*** TEMPERATURE AT INDICATED ENTHALPY CANNOT BE FOUND)
   ANS=TEMm
   RETURN
*** BUBTP ***

ENTRY BUBTP
ICAL=1
ASSIGN 60 TO LOC
GO TO 1000

ANS=0.
WRITE(*,611)
FORMAT(* *** ABOVE CRITICAL PRESSURE - DEW/BUBBLE POINT CANNOT BE FOUND)
RETURN

IF(ICAL.EQ.2) GO TO 651

***
CALC APPROX. STARTING VALUE FOR TEMTUR
TEMTUR=TEMST(AAMW,PRSSUR)-15.
FRDV=0.07
***

IF(COUNT.EQ.1) GO TO 70

LOS=1
GO TO 4000

IF(PRESSUR.GT.(1.05*PCRT)) 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

ASSIGN 65 TO LOC
GO TO 11000

DO 66 I=NFC,NLC
TEMP=NEWX(I)
NEWX(I)=X(I)

X(I)= TEMP
COUN=0

FFLAG=.FALSE.
IF(COUNT.EQ.1) GO TO 68

LOS=2
GO TO 8000

DO 69 I=NFC,NLC

TEMP=KV(I)*NEWX(I)

IF(ABS((TEMP-X(I))/TEMP).GT.1.E-5) FFLAG=.TRUE.

DO 160 I=NFC,NLC

TEMP=KV(I)*NEWX(I)

X(I)= NEWX(I)

IF(ABS(SUMM)/TMOLE.GT.5.E-3) GO TO 163

CONTINUE

DO 162 I=NFC,NLC

LIST(I)=KV(I)

IF(ICAL.EQ.2) GO TO IPR,(310,410)
ANS=TEMSTUR
RETURN

ASSIGN 164 TO LOC
GO TO 2000

CONTINUE

WRITE(*,6166)
FORMAT(* BUBBLE POINT CANNOT BE DETERMINED)

ENTRY DEWTP
ICAL=1
ASSIGN 70 TO LOC
GO TO 1000

*** DEWTP ***
70 LOS=2
IF (ICAL.EQ.2) GO TO 751
***
C CALC APPROX STARTING VALUE FOR TEMTUR
TEMST=TEMST(AAMW,PRSSUR)
FRDV=0.07
***
751 GO TO 4000
73 IF (PRSSUR.GT.(1.05*PCRIT)) GO TO 61
TT=TEMSTUR
ENACT=0.
CALL ZEROILNACT,NOCOMP
CALL ZEROILNEWX,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)/KVI(I)
IF (ABS((TEMP-X(I))/TEMP).GT. 1.0.E-5 ) FFLAG=.TRUE.
170 X(I)=TEMP
COUN=COUN+1
IF (COUN.LT.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+(1.0/KVI(I)-1.0)*X(I)
171 CONTINUE
IF (ABS(SUMM)/TMOLE .GT. 5.0.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 ENTRY PBUB
ICAL=2
FRDV=-0.25
ASSIGN 31U TO IPR
GO TO 650
310 ANS=PRSSUR
RETURN
C C ENTRY PDEW
ICAL=2
FRDV=-0.25
ASSIGN 410 TO IPR
GO TO 750
410 ANS=PRSSUR
RETURN
C C INTERNAL FUNCTION
C C 1000 IF(ARG.GT.0) GO TO 1007

183
J=-ARG
DO 1006 I=1,NOCOMP
1006 X(I)=XOUT(I,J)
TEMPUR=OUT(J)
PRSSUR=OUT(J)
HCOUNT=OUT(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)
TEMPUR=IN(J)
PRSSUR=PIN(J)
HCOUNT=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=1
GO TO 1005
1004 NLC=1
1005 CONTINUE
NCMP=COUNT
GO TO LOC.((20,30,40,50,60,70)

INTERNAL FUNCTION

ICAL=1 ITERATING ON TEMTUR
ICAL=2 ITERATING ON PRSSUR

2000 IF(ICAL.EQ.1) VAR=TEMTUR
IF(ICAL.EQ.2) VAR=PRSSUR
OLVAR=VAR
IF(COUNT.GT.1) GO TO 2001

SUMM + TEMTUR TOO LOW, PRSSUR TOO HIGH
- TEMTUR TOO HIGH, PRSSUR TOO LOW

2001 IF(SUMM)05,2005,2003
2002 IF(II.EQ.2) 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
2008 IF(SUMM.GT.0.) GO TO 2010
VHIGH=VAR
SUMH=SUMM
GO TO 2012
2010 VLOW=VAR
SUML=SUMM
2012 VAR=(VLOW*SUMH-VHIGH*SUML)/(SUMH-SUML)
DIFV=VHIGH-VLOW
IF(NCMP.EQ.1) GO TO 2113
SIGN=SIGN(1.*FRDV)
DIFV=DIFV*SIGN
IF(ICAL.EQ.1) VIOL=2.
IF(ICAL.EQ.2) VIOL=0.02*VAR
IF(DIFV.GT.VTOL) GO TO 2113

C** SPECIAL RESTART PROCEDURE FOR BUBTP + PBUB CONVERGENCE
RE-ESTABLISHES LOW TEMP OR HIGH PRES (LOW K VALUE) LIMIT
I1=-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
IF(ICAL.EQ.2) PRESSUR=VAR
IF(PRESSUR.GT.(1.05*PCRIT)) GO TO 61
GO TO LOC(56,164,174)

C INTERNAL FUNCTION ***** CALCAB *****

3000 A=B=SUMX=0.
DO 3001 I=NFC,NLC
B=B+BASEA(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(4001,8001,14001,2274)

C INTERNAL FUNCTION ***** PCRIT *****

4000 ASSIGN 4001 TO LOC
GO TO 3000
4001 TRASH=(4.94/ASQDB)**.6666667
PCRIT=.0867/(TRASH*B)
GO TO (63,73),LOS

C INTERNAL FUNCTION ***** ZFAC *****

ORIGINAL NEWTON-RAPHSON ITERATIVE SOLUTION FOR REDLICH-KWONG
HAS BEEN REPLACED WITH ANALYTICAL SOLUTION FOR CUBIC IN Z
THE VAPOR COMPRESSIBILITY ZFAC IS ALWAYS THE FIRST ROOT-ZZ

5000 BP=B*PRESSUR
ZB1=-1.
ZB2=BP*(ASQDB-1.0-BP)
ZB3=-ASQDB*BP*BP
ZB10V3=ZB1/3.0
ZALF=ZB2-ZB1*ZB10V3
ZBET=2.*G*ZB10V3**3-ZB2*ZB10V3+ZB3
ZBETOVB2=ZBET/2.
ZALFOV3=ZALF/3.
ZCUAOV3=ZALFOV3**3
ZSQBOV2=ZBETOVB2**2
ZDEI=ZSQBOV2+ZCUAOV3
C FOR ZDEI +VE THERE IS ONLY ONE REAL ROOT
FOR ZDEL -VE THERE ARE THREE REAL ROOTS

IF(ZDEL)5003,5005,5004

ZEPS=SQRT(ZDEL)
ZRCU=-ZBTOV2+ZEPS
ZSCU=-ZBTOV2-ZEPS
ZSIR=1.0
ZSIS=1.0
IF(ZRCU)5007,5008,5009

5007 ZSIR=-1.0
5008 IF(ZSCU)5009,5010,5010

5009 ZSZ=ZSIR*(ZSIR*ZRCU)**0.33333333
      ZSZ=ZSIS*(ZSIS*ZSCU)**0.33333333
      ZZ=ZSZ+ZSZ-ZB10V3
      GO TO 5100

5003 ZQUOT=ZSBOV2/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)-ZB10V3

5100 ZFACTR=ZZ
ZFACT=ZFACTR
H=BP/ZFACT
GO TO LOC(23,8002,14002,75)

INTERNAL FUNCTION ****** LIQDEN ******

DO 6101 I=NFC,NLC
PST=PST+X(I)*ATC(I)/TMOLE
PSV=PSV+X(I)*AVC(I)/TMOLE
6101 ZCE=ZCE+X(I)**2*ZC(I)/TMOLE
PS=(ZCE*1.0**3.3E*PST)/PSV

ACON=DPOLY(17.41425E0,-214.578E0,989.625E0,-1522.06E0,ZCE)
IF(ZCE)6120,6111,6111
IF(ZCE)6104,6102,6102

6111 BCON=DPOLY(6.2091E0,-402.063E0,501.0E0,641.0E0,ZCE)
6120 DCON=93E0-BCON
TROD=TEMTR/PST
ARED=1.0E0-TROD
IF(TROD.GE.1.0E0) ARED=0.0E0
SRED1=ARED**1.0/3.0
SRED2=SRED1*SRED1
SRED4=SRED2*SRED2
RHORS=DADDY(1.0E0,ACON,BCON,0.0E0,DCON)
E27=DADDY(1.0E0,1.0E0,-.626E0,-.66E0,3.699E0,-2.198E0)
IF(TROD.GE.1.0E0) GO TO 6300
F27=-ALOG(TROD)
G27=.268E0*(1.0E0+F27)**2.0967*(1.0E0+F27)**2.0967
H27=1.41E0*(1.0E0-TROD)**.75*EXP(-7.848E0*(1.0E0-TROD))
GO TO 6301

3600 F27=.268E0*(TROD**2.0967)
G27=.268E0*(TROD**2.0967)
H27=.268E0*(TROD**2.0967)

3600 IF(ZCE)>10.0E0,45.22E0,-103.79E0,114.44E0,-47.38E0)
IF(ZCE)6104,6102,6102
IF(ZCE)6104,6102,6102
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)>23E0) GO TO 6105
DELPZ=ZCE-23E0/.005E0
AFAC=3.15E0+DPOLY(-28339E-2,-358331E-2,-31658E-2,.416557E-3)
1 DELPZ*DELPZ
GO TO 6103
**AFAC=3.15E0**

**PRS=EXP(2.302585E0*AFAC*(1.E0-(1.E0/TROD)))**

**DELP=(PRS5SUR/PSP)-PRS**

**TT=DELP**

**IF(DELP.*LT.*2E0) TT=0.2E0**

**DELDUM=EXP(27*FRK1*ALOG(TT)+G27*EXP(H27*TT)**

**IF(DELP.*LT.*2E0) DELDUM=DELDUM*(DELP/2E0)**

**IF(ABS(ZCE.-27E0).GT.1.E-10) GO TO 6107**

**RDZC=0.**

**GO TO 6112**

**J=3**

**IF(ZCE.GT.27E0) J=1**

**IF(ZCE.LT.-27E0) AND ZCE.GT.-26E0) J=2**

**RJ=MADDY(FRJ(1,J),FRJ(2,J),FRJ(3,J),FRI(4,J),FRI(5,J))**

**RK1=MADDY(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=RJ1+RJ1*ALOG(TT)+RK1*EXP(RK1*TT)**

**IF(DELP.*LT.*20E0) RDZC=RDZC*(DELP/2E0)**

**RO=ROCRIT+DELDUM+RDZC**

**ROCRIT=(PSP*AAMW)/ZCE*10.73E0*PST**

**RHO=(PRSSUR*AAMW)/(10.73*TEMURT*RO)**

**ZLIQ=(PRSSUR*AAMW)/(10.73*TEMURT*RO)**

**GO TO LOC,(25,14004)**

**INTERNAL FUNCTION ****** LIQPRM ********

**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**

**INTERNAL FUNCTION ****** VAPPRM ********

**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**

**INTERNAL FUNCTION ****** EQR ********

**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)**

**INTERNAL FUNCTION ****** GASFUG ********

**10000 ZT=ZFCCTOR -1.**

**ENPHI=0.**

**IF(H.*,GT.*999E0) GO TO 10002**

**ENPHI=ALOG(ZFCCTOR*(1.-H)).**

**BT=BT*TEMURT**

**AT=AT*TEMURT*1.25/2.**

**DO 10001 I=NFC,NLC**

**BTT=BASEB(I)/BT**

**10001 LNPHI(I)=ZT*BTI-ENPHI-ASCON*(BASEA(I)/AT-BTT)**

**GO TO LOC,(76,8003)**

**INTERNAL FUNCTION ****** LIQACT ********

**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
LNACT(I) = AV25(I) * (ADEL(I) - SUMDEL)**2 / TEMTUR / 1.1033
GO TO LOC(*65,7002)
C C 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. 0.03 .AND. J .EQ. 3) J = 2
ENNU = ((COEFFT(J, 5) * TRED + COEFFT(J, 4) * TRED + COEFFT(J, 3) * TRED +
1 * COEFFT(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 -
3 * ALOG(PRED)) / 2.302585
C THIS VARIATION SUGGESTED BY GRAYSON AND STREED.
12001 ENNU = ENNU + AOMEG(I) * ((-3.15224 * TRED * TRED + 8.65808 * TRED -
4.23893 / TRED - 1.2206 / TRED - 0.611)
GO TO LOC(*64,7001)
C C 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)
SDTA = SDTA + X(I) * DTA(I)
13001 TOK = TRED / 1.8
HBASE = ((SDTA / 4.2 * TOK + SGAM / 3.1 * TOK + SBET / 2.8 * TOK + SAPH) * TRE
GO TO LOC(*14008)
C C 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, ZFACT0) * 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(174H)** ENTHALPY CANNOT BE CALCULATED, VAPOR FRACTION IS IN
1 PROPERLY SPECIFIED)
GETH = 0.
14009 GO TO (33, 52) * LOS
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)

***** EQUIP TYPES ADBF 10
      MIXR 21
FLASH ROUTINE - 3 ENTRIES
      ADBF - ADIABATIC FLASH
      ISOF - ISOTHERMAL FLASH
      MIXR - ADIABATIC MIXING
PRESSURE CHANGE ALLOWED FOR ADBF + MIXR ONLY
PEX=0. INDICATES OUTLET PRESSURE (PEX) IS PIN(1)

EQUIP OUTPUT VECTOR CODING -
      1. - 2. EQUIP NO + TYPE
      3. - 6. INLET/OUTLET STREAM NOS
      7. OUTLET VAPOR FRACTION (*100)
      8. - 9. (1ST) INLET/OUTLET PRESSURES (PSIA)

***** COMMON DECK
COMMON/CONT(NE),NIN,NOUT,NOCOMP
COMMON/EQUIP/EQUIP(15)
COMMON/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),XINC(8,4)
COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),TMOUT(4),XOUT(8,4)

*****
LOGICAL FLAG, FFAG
DATA FLAG/.FALSE./
REAL DUM(1), NEWDF, KS(8), OLDKS(8), OEQ(8), EQR(8)
INTEGER TCNT, COUNT

***** ENTRY ADBF
ETOYPE=10.
ICAL=0
XHIN=HIN(1)
GO TO 100

***** ENTRY ISOF
ICAL=1
GO TO 100

***** ENTRY MIXR
ETOYPE=21.
IMIX=1
ICAL=0
XH=XTM=0.
CALL ZERO(XOUT,1,NOCOMP)
DO 121 J=1,NIN
   XH=XH+HINC(J)
   XTM=XTM+TMIN(J)
DO 121 I=1,NOCOMP
121 XOUT(1,1)=XOUT(1,1)+XIN(I,J)
  HIN(1)=XH
  XHIN=HIN(1)
  TMIN(1)=XTM
DO 122 I=1,NOCOMP
122 XIN(1,1)=XOUT(1,1)
GO TO 104

100 IMIX=0

104 TCNT=1
KCNT=0
CALL ZERO(BPOUT,60)
CALL ZERO(OLDKS,24)
TSAV=TEMP=TIN(1)
PSAV=PIN(1)
VSAV=VFIN(1)
BBT=BPINC(1)
DWT=DPINC(1)

PX=PEX
IF(PX.EQ.0.) PX=PIN(1)
IPEX=0
IF(IMIX.EQ.1.OR.PX.NE.PIN(1)) IPEX=1
C

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
NL = NL - 1
GO TO 102
101 NL = 1
102 CONTINUE
IF(ICAL.EQ.0) GO TO 4

C

TOUT(2) = TOUT(1) = TT = TEMP
TMOUT(2) = TMOUT(1) = TMIN(1)
DO 3 I = NF, NL
3 XOUT(1,2) = 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
MAKE INITIAL K-VALUE ESTIMATES - AT FEED D.P.
CALL DEWTMP(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 = VFOUT(1)
II = 1.
GO TO 9

C

MUST CALC DEW + BUBBLE POINTS OF INPUT FOR PRES CHANGE OR MIXING

4 IF(IP EX.EQ.0) GO TO 6
CALL BUBTP(1, BBT, EOR)
IF(COUNT.GT.1) GO TO 5
DWT = BBT
GO TO 6
5 CALL DEWTMP(1, DWT, KS)
6 TIN(1) = DWT
VFOUT(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
VFOUT(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

**** FOR ONE COMPONENT SYSTEM, CALCULATE V DIRECTLY

TOUT(2)=TOUT(1)=DWT
V=(HIN(I)-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 HOUT(1)=HIN(I)*V
HOUT(2)=HIN(I)-HOUT(1)
GO TO 150

28 TNEG=BBT
TPOS=DWT
AA=ABS(HIN(1))
FNEG=(HL-HIN(1))/AA
FPOS=(HV-HIN(1))/AA
TEMP=(DWT*FNEG-BBT*TPOS)/(FNEG-FPOS)
CALL ZEROCE(QR,NOCOMP)
GO TO 2

10 FLAG=.TRUE.
DO 32 I=NF+NL
OLDIF=EOR(I)-OLEQ(I)
NEWDIF=KS(I)-OLDKS(I)
IF((ABS(NEWDIF)/KS(I))*1.E-4) GT .5) FLAG=.FALSE.
OLDKS(I)=KS(I)
IF(KCNT LE 2) GO TO 33
IF((ABS(NEWDIF)/KS(I))*LT.1E-4) GO TO 33
QWEG=1.-OLDIF/NEWDIF
IF(ABS(QWEG)*LT.1E-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 EOR(I)=KS(I)
IF(FLAG OR KCNT GT 10) GO TO 48

V=.5
DO 42 COUNT=1,10
SUM=DSUM=0.
DO 43 I=NF+NL
TEMPO=(KS(I)-1.)/(V*(KS(I)-1.))
TEMPA=TEMPO*XIN(I,1)
DSUM=DSUM+TEMPO*TEMPO
43 SUM=SUM+TEMPA
TEMPA=SUM/DSUM
IF(ABS(TEMPA) LT.1E-4) GO TO 45
V=V+TEMPA
42 IF(V GT.1) V=0.99
42 IF(V LE 0) V=0.01
45 VF(a)=V
SUM=DSUM=0.
DO 47 I=NF+NL
TEMPO=(1.-V)*XIN(I,1)/(V*(KS(I)-1.))
IF(TEMPO LT.0) TEMPA=0.
TEMPO=XIN(I,1)-TEMPA
IF(TEMPO LT 0.) GO TO 46
TEMPA=TEMPO*XIN(I,1)
TEMPO=0.
46 DSUM=DSUM+TEMPO
SUM=SUM+TEMPA
XOUT(1,2)=TEMPA
47 XOUT(I,1)=TEMPO
TMOUT(1)=DSUM
TMOUT(2)=SUM

11 CALL KVAL(-2*ADKS)
CALL KVAL(-1*ADKS)
KCNT=KCNT+1
GO TO 10

48 CALL ZERO(HOUT,2)
DO 52 J=1,2
52 IF((TMOUT(J)/TMIN(J)) .GT. 0.001) CALL ENTH(-J,HOUT(J),SUM)
IF(ICAL.EQ.1) GO TO 53
FTEMP=(HOUT(1)+HOUT(2)-HHIN(1))/TMOUT(1)
IF(ABS(FTEMP) .LT. 25) GO TO 53
IF(TCNT .LT. 10) GO TO 55
WRITE(*,656)

56 FORMAT(* FLASH DID NOT CONVERGE*)
GO TO 53

55 IF(FTEMP .GT. 0.1) GO TO 73
TNEG=FTEMP
FNEG=FTEMP
GO TO 74

73 TPOS=FTEMP
FPOS=FTEMP

74 TT=(TPOS*FNEG-TNEG*FPOS)/(FNEG-FPOS)
IF(ABS(TT-TTEMP) .LT. 0.05) GO TO 53
FF=ABS(FNEG/FPOS)
DTF=TPOS-TNEG
IF(FF .LT. 10) TT=TNEG+0.7*DTF
IF(FF .LT. 1.1) TT=TNEG+0.3*DTF

60 TEMP=TT
TCNT=TCNT+1
TOUT(2)=TOUT(1)=TIN(1)=TT
KCNT=0
CALL ZERO(EQR,NOCOMP)
GO TO 11

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

66 V=TMOUT(1)/(TMOUT(1)+TMOUT(2))
XHIN=HOUT(1)+HOUT(2)
II=1
GO TO 9

** 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)

150 TIN(1)=TSAV
PIN(1)=PSAV
VFIN(1)=VSAV

CALCULATE BUBBLE, DEW POINTS FOR OUTLETS

JJ=1
IF(NOUT .EQ. 1 .OR. TMOUT(2) .EQ. 0.) GO TO 154
IF(TMOUT(1) .GT. V) GO TO 151
JJ=2
GO TO 154

151 DO 152 JJ=1,2
CALL BUBTP(-J, BOUT(J), KS)
CALL DEWTP(-J, DOUT(J), KS)
152 CONTINUE
GO TO 155
154 BOUT(JJ) = BBT
DOUT(JJ) = DWT
155 IF (ICAL.EQ.1) RETURN

C** SET OUTPUT PARAMETERS
C
CALL ZERO(EQUIP, 15)
EQUIP(2) = EQTYPE
EQUIP(7) = 100.*V
EQUIP(8) = PSAV
EQUIP(9) = PX
RETURN
END
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 specified 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.

\[
\frac{T_{\text{OUT}}}{T_{\text{IN}}} = \left( \frac{P_{\text{OUT}}}{P_{\text{IN}}} \right)^{\frac{n-1}{n}}
\]

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

\[
W = \Delta H
\]

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

*** EQUIP TYPE 100

COLUMN DESIGN MODEL
BASED ON HENGSTEBECKS PSEUDO-BINARY PROCEDURE
COMPUTES COLUMN SIZE + COST FOR SPECIFIED KEY SPLIT
FOR 4IN BUBBLE CAPS

EQUIP INPUT VECTOR CODING -
1. - 2. EQUIP NO + TYPE
3. CONDENSER CONFIGURATION FLAG - 0. TOTAL CONDENSER,
   1. PARTIAL CONDENSER
   -1. 0/H PRODUCT WITHDRAWN AS VAPOR BEFORE TOTAL CONDENSER
4. REFLUX RATIO FACTOR - R/RMIN
5. - 6. OVERHEAD LIGHT + HEAVY KEY MOLE FRACTIONS
7. - 8. BOTTOM LIGHT + HEAVY KEY MOLE FRACTIONS
9. -10. LIGHT + HEAVY KEY COMPONENT NOS
11. COLUMN PRESSURE (PSIA)
12. TRAY SPACING (INS)

PARAMETERS -
STRES - STRESS CARBON STEEL - PSI
TREFF - TRAY EFFICIENCY - FRACTION
ACOL, BCOL - COST COEFFS - COLUMN
ATR, BTR - COST COEFFS - TRAYS (SS BUBBLE TRAYS)
CFCOL - FACTOR TOT CAP INV/COL CAP COST
AMORT - FRACTIONAL CHARGE ON CAPITAL/YR
CORR - CORROSION ALLOWANCE FOR SHELL - INS

---------------------------------------------
MATERIALS DATA - NOT IMPLEMENTED BY PROGRAM

<table>
<thead>
<tr>
<th>STEEL</th>
<th>TEMP RANGE</th>
<th>STRESS</th>
<th>COST FACTOR</th>
<th>STRESS CORR</th>
<th>COST FACTOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CARBON</td>
<td>TO -50F</td>
<td>13750</td>
<td>1.00</td>
<td>-</td>
<td>1.00</td>
</tr>
<tr>
<td>NICKEL</td>
<td>TO -150F</td>
<td>16000</td>
<td>2.90</td>
<td>-</td>
<td>1.72</td>
</tr>
<tr>
<td>STAINLESS</td>
<td>BELOW -150F</td>
<td>18750</td>
<td>3.5</td>
<td>-</td>
<td>2.56</td>
</tr>
</tbody>
</table>

OUTPUTS -
D 1. 0/H PRODUCT - LIQUID (PCOND=0.), VAPOR (PCOND=-1.,1.)
B 2. BOTTOMS PRODUCT - LIQUID
 V 3. FLOW TO CONDENSER (PARTIAL OR TOTAL) - VAPOR
  VBAR 4. FLOW TO REBOILER - LIQUID

EQUIP OUTPUT VECTOR CODING -
1. - 2. EQUIP NO + TYPE
4. CONDENSER CONFIGURATION FLAG (PCOND)
5. OUTLET TEMP FOR PARTIAL CONDENSER (DEG R)
6. REFLUX RATIO
7. NO OF RECTIFYING TRAYS
8. NO OF STRIPPING TRAYS
9. COLUMN DIAMETER (FT)
10. COLUMN THICKNESS (IN)
11. MATERIAL COST FACTOR
12. COLUMN SHELL WEIGHT (LBS)
13. CAPITAL COST ($)  
14. OPERATING COST ($/YR)
15. TOTAL COST ($/YR)

**** COMMON DECK
COMMONT/CONTL/NE,NIN,NOUT,NOCOMP
COMMONT/EMI/EMI(15,10)
COMMONT/EQUIP/EQUIP(15)
COMMONT/SIN/BPIN(4),DPIN(4),TIN(4),PIN(4),HIN(4),VFIN(4),TMIN(4),
1 XIN(8,4)
COMMONT/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4),
1 XOOUT(4),XOUT(8,4)
COMMONT/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APP,APRR,
1 ARR, TRRR

****
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))

DATA STRES,TREFF,ACOL,BCOL,ATR,BTR,CFCOL,CORR/
13750.,0.7,14.5,0.7,48.,1.7,40.0625/

QUAD(A,B,C)=(SQR(T**2-A*C))/2*A)

SET INPUT PARAMETERS

PCOND=EMIC3,NEJ,RFAC=-EMIC4,NEJ,YL=EMIC5,NEJ,YH=EMICH1,NEJ,XL=EMIC7,NEJ,
XH=EMIC8,NEJ,NLK=EMIC9,NEJ,NHK=EMIC10,NEJ,NF=EMIC11,NEJ,
PRES=EMIC12,NEJ,
PSAV=PINC11
POUT(4)=POUT(3)=POUT(2)=POUT(1)=PIN(1)=PRES
VFOUT(3)=VFOUT(1)=1.
VFOUT(4)=VFOUT(2)=0.
CALL ZEROTOP,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(1)/MIN(1)
IF(XX.LT.0.002) GO TO 4
IF(NL.GT.0.02) GO TO 2
NF=NL=1
GO TO 4
2 NL=I
4 CONTINUE

****

CALCULATE Q VALUE(EQUIV. TO VAP. FRAC.) FOR ACTUAL FEED
USE THIS VALUE FOR THE PSEUDO-BINARY SYSTEM

VFSAV=VFINC1 I
TSAV=TIN(I)
IF(PRES.NE.PSAV) CALL BUBTPB,BPIN(I),TKV)
TIN(1)=BPIN(1)
VFIN(1)=0.
CALL ENTH(1,HV,DUM)
CALL DEWTP(1,DPINC11,XKV)
TIN(1)=DPINC11
VFIN(1)=1.
CALL ENTH(1,HV,DUM)
VFIN(1)=VFSAV
TIN(1)=TSAV
QQ=(HIN(1)-HL)/(HV-HL)

C

CALCULATE KEY MOLE FLOWS CORRESPONDING TO GIVEN MOLE FRACTIONS
B=FEED(NLK)-YL*FEED(NHK)/YH/(XL-XH*YL/YH)
D=MIN(NL-8
TOP(NLK)=D*YL
TOP(NHK)=D*YH
BOT(NLK)=B*XL
BOT(NHK)=B*XH

C

COMPUTE ALPHA VALUES - CORR TO FEED AT DEW POINT
DO 6 I=NF,NL
6 ALPHA(I)=XKV(I)/XKV(NHK)
200

SET UP LINEAR LN(D/B) VS LN(ALPHA) RELATION
SLOPE=(ALOG(TOP(NLK)/BOT(NLK))-ALOG(TOP(NHK)/BOT(NHK)))/(ALOG(ALPHA(NLK))-ALOG(ALPHA(NHK)))
XINT=ALOG(TOP(NHK)/BOT(NHK))-(ALOG(ALPHA(NHK)))*SLOPE

CALCULATE LN(D/B)'S FOR ALL COMPS FROM THEIR ALPHA'S
CALCULATE B'S + D'S FOR ALL NON KEYS

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)
    B=B+BOT(I)
8 TMOUT(1)=D
    TMOUT(2)=B

CALCULATE CRITICAL LN(D/B)'S
    DBCL=DBLN(NLK)+7*(DBLN(NLK)-DBLN(NHK))
    DBCH=DBLN(NHK)-7*(DBLN(NLK)-DBLN(NHK))

CALCULATE KEY PORTIONS FOR LIGHT NON KEYS + SUM FOR EFFECTIVE LK

BINF =FEED(NLK)
BINDL=TOP(NLK)
BINBL=BOT(NLK)
II=NLK-1
IF (II+I,NF) GO TO 15
DO 14 I=NF,II
    IF (DBLN(I)-DBCL) 10,10,11
    10 - IF LN(D/B) IS LESS THAN CRIT VALUE INCLUDE WHOLLY IN LK
    11 - IF GREATER CALCULATE PORTION TO BE INCLUDED
    A1=FEED(I)
    A2=TOP(I)
    A3=BOT(I)
    GO TO 12
11 A2=BOT(I)*EXP(DBCL)
    A3=0.
    A1=A2
12 BINF=BINF+A1
    BINDL=BINDL+A2
    BINBL=BINBL+A3
14 REPEAT FOR HEAVY NON KEYS SIMILARLY

BINFH=FEED(NHK)
BINDH=TOP(NHK)
BINBH=BOT(NHK)
II=NHK+1
IF (II+I,NF) GO TO 21
DO 20 I=II,NF
    IF (DBCH-DBLN(I)) 17,17,18
    17 - IF LN(D/B) IS GREATER THAN CRIT VALUE INCLUDE WHOLLY IN HK
    18 - IF LESS CALCULATE PORTION TO BE INCLUDED
    A1=FEED(I)
    A2=TOP(I)
    A3=BOT(I)
    GO TO 19
18 A2=0.
    A3=TOP(I)*EXP(-DBCH)
    A1=A3
19 BINF=BINFH+A1
    BINDH=BINDH+A2
    BINBH=BINBH+A3
20 REPEAT FOR HEAVY NON KEYS SIMILARLY

CALCULATE PARAMETERS FOR EFFECTIVE BINARY

XF=BINF/(BINF+BINFH)
XD=BINDL/(BINDL+BINDH)
XW=BINBL/(BINBL+BINBH)
R1=ALOG(BINDL/6*INBL)
R2=ALOG(BINDH/BINBH)
RV1 = EXP((R1-XINT1)/SLOPE1)
RV2 = EXP((R2-XINT1)/SLOPE1)
RV = RV1/RV2
RV is the normalised rel. vol. of the effective LK
Q1 = 1.0 - QQ

Now apply Stoppels calculation for theoretical stages

Compute minimum reflux ratio - find feed/equil. intersection

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.0.0 OR XX1.GT.1.0) XMIN = XX2
YMIN = SS*XMIN+XNT
SSMIN = (XD-YMIN)/(XD-XMIN)
RMIN = SSMIN/(1.0-SSMIN)
RR = RMIN*RFAC
R1 = RR + 1.0
R11 = 1.0/R1

Calculate intersections

Feed line + rect ol
XQ = (XF/QQ-XD/R1)/(RR/R1+Q1/QQ)
YO = XQ*RR*R11+XD*R11

Equil line + rect ol
SS = RR*R11
XNT = R11*XD
Assign 94 to KINT
GO to 96
94 XE = AA + QU
XD = AA - QU
YE = SS*XE+XNT
YO = SS*XD+XNT

Equil line + strip ol
SS = (YO-XW)/(XQ-XW)
XNT = XW*(1.0-SS)
Assign 95 to KINT
GO to 96
95 XED = AA + QU
XOD = AA - QU
YEDE = SS*XED+XNT
YODE = SS*XOD+XNT

Calculate TR TS
TR = ALOG(((XD-YO)*(XE-XO))/((YE-XD)*(XQ-XO)))/ALOG((YO*XE)/(XO*YE))
TS = ALOG(((YQ-YOD)*(XED-XW))/((YED-YQ)*(XW-XOD)))/ALOG((YOD*XED)/(X

Quit

Set condenser + reboiler flows
CDFACT = RR + 1.0
IF(PCOND.EQ.-1.0) CDFACT = RR
RBFAC = (RR*D+TMIN(1.0-QQ)-B)/B
IF(PCOND.LE.0.0) GO to 109

Compute partial condenser
Top are vapor flows from P.C.
Compute liquid flows from P.C. + vapor flows to P.C.
Use output vector 4 as temp location for liquid flows

Compute K-values for O/H at dew point
CALL DEWTP(-1.0,DWT,XKV)
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

109 TMOUT(3)=D*CFDFACT
DO 111 I=NF,NL
111 XOUT(I,3)=CFDFACT*TOP(I)
112 TMOUT(4)=RBFAC*B
DO 113 I=NF,NL
113 XOUT(I,4)=RBFAC*BOT(I)

C
C SUBTRACT 1 TRAY FOR PARTIAL CONDENSER
IF(PCOND.EQ.1.0) TR=TR-1.
C
C ROUND TRAY NUMBERS
TR=TR/MTREFF
II=TR+1.
TR=II
TS=TS/MTREFF
II=TS+1.
TS=II

C
C SET OUTPUT BUBBLE/DEW POINTS, TEMPS + ENTHALPIES
C
DO 114 J=1,3
CALL BUBTP(-J,BPOUT(J),XKV)
114 CALL DEWT (-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)*RBFAC

C*
C CALCULATE COLUMN DIAMETER FOR 4IN BUBBLE CAPS
C COMPUTE DENSITIES + FLOWS
C
DO 122 I=1,2
J=I+2
VFSAV=VFOUT(J)
IF(PCOND.EQ.-1.0.AND.J.EQ.3) VFLOW=VFLOW*((RR+1.0)/RR)
VFOUT(J)=1.
CALL DENS(-J,RO,DUM)
VFLOW=TMOUT(J)/RO
ROV(I)=XMW*RO
IF(PCOND.LE.0.0.OR.I.EQ.2) GO TO 120
ROL(1)=ROPC
GO TO 121
120 VFOUT(J)=0.
CALL DENS(-J,RO,DUM)
ROL(1)=XMW*RO
121 AREA(I)=0.0155*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
C
THICK=(PRES*DINS)/(1.6*STRES-1.2*PRES)+CORR
WT=((TR+TS+3.0)*TSPACE+DINS)*3.14*DINS*THICK*0.283
CAP=(ACOL*(WT)**BCOL+(TR+TS)*ATR*(DCOL)**BTR)*CFCOL
CJR=CAP*AMORT
SET OUTPUT PARAMETERS
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

*******
INTERNAL FUNCTION TO CALC. EQUIL. + OP. LINE INTERSECTIONS

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)

***** EQUIP TYPES (IOP=) 1,2

HEAT EXCHANGER DESIGN ROUTINE
COMPUTES AREA + YEARLY COST FOR 1-1 COUNTERFLOW EXCHANGER

NOMENCLATURE -
IOP - OPERATION TYPE (ON FIRST INPUT)
  1 COOLING
  2 HEATING
IS2 - 2ND STREAM TYPE
  1 STEAM
  2 WATER
  3 REFRIGERANT (PROPANE OR ETHYLENE)
  4 PROCESS STREAM
  5 STREAM 1 PSEUDO-SERVICE (PARALLEL PROC)
  6 STREAM 2 PSEUDO-SERVICE (PARALLEL PROC)
  (IPS IS PSEUDO-SERVICE STREAM NO - IF GT 0)
TEX(1) - SPEC EXIT TEMP FOR STREAM 1
  (IF 0, EXCHANGE TO APPROACH)
  (FOR IS2=4 COMPUTE TO SAT TSPEC FOR LIMITING STREAM (MIN Q))
TEX(2) - IS2=1,2 N.A.
  IS2=3 REFR EVAP TEMP
  IS2=4 SPEC EXIT TEMP FOR STREAM 2
Q - HEAT TRANSFERRED TO 1ST STREAM

PARAMETERS -
HRS - NO. OF OPERATING HRS/YR
AEX,BEX - EXCHANGER COST COEFFS.
AMORT - FRACTION OF CAPITAL CHARGED/YR
TWAT - COOLING WATER TEMP
DTW - WATER TEMP RISE
CWAT - COOLING WATER COST
APPR - MIN TEMP APPROACH (GENERAL)
APPR - MIN TEMP APPROACH (REFR IS2=3,4,5)
APPR - MIN TEMP APPROACH (REFR IS2=3,4,5 WITH TLT.TRRR)
TS - STEAM COND. TEMP
HVS - STEAM LATENT HEAT
CS - STEAM COST
TC - LOW TEMP LIMITS FOR CARBON, NICKEL STEELS
CFEX - COST FACTORS FOR NICKEL, STAINLESS STEELS

PROCESS FILM COEFFICIENTS -
  COND OR REB
  COOLING OR HEATING PROCESS LIQUIDS
  COOLING OR HEATING PROCESS VAPORS - XU1
  (PR GT PLOW) - XU2
  (PR LT PLOW) - XU4

EQUIP OUTPUT VECTOR CODING -
  1 - 2. EQUIP NO + TYPE
  3 - 6. INLET/OUTLET STREAM NOS
  7. EXCHANGER HEAT LOAD + FOR HEATING 1ST STREAM (BTU/HR)
  8. TRANSFER AREA (FT 2)
  9. ENTROPY INCREASE/BTU (/DEG R *10**5)
  10. SERVICE OR PSEUDO-SERVICE FLOW (MOLES/HR)
  11. REFRIGERANT TEMP LEVEL OR PSEUDO-SERVICE OUTLET TEMP
  12. SPEC (DEG R)
  13. CAPITAL COST (S)
  14. OPERATING (SERVICE) COST ($/YR)
  15. TOTAL COST ($/YR)

***** 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)
  XIN(8,4)
COMMON/SOUT/BPOUT(4),DPOUT(4),TOUT(4),POUT(4),HOUT(4),VFOUT(4)
  TMOUT(4),XOUT(8,4)
COMMON/PARAM/AMORT,HRS,TWAT,DTW,CWAT,TS,HVS,CS,CKWH,APPP,APPR,
**ARRR** TRRR

**C****

**DIMENSION SIDUM(4,7), SODUM(4,7)**
**EQUIVALENCE (SIDUM,APIN),(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)**

**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/**

**WRITE(6,600)TEX,IOP,IS2**
**FORMAT(//3H ** • EXCHANGER - TSPECS*,2F8.1,* IOP,IS2*,2I4/)**

**TSPEC(1)=TEX(1)**
**TSPEC(2)=TEX(2)**
**SIGNO=(3-2*IOP)**
**IPS=IS2-4**
**FM=1.**
**AP=APPP**
**IF(IS2.GT.2) AP=ARRR**
**IF(TSPEC(J).GT.0 .AND.TSPEC(J).LT.TRRR) AP=ARRR**

**SET TEMPS ETC FOR IS2=1,2,3**

**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.**
**GO TO 30**

**30 TOUT(2)=TIN(2)+DT2**

**C**

**FIND LIMITING STREAM (MIN Q)**

**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)**

**IF(RABS(QS(1)/QS(2)))**
**IF(RABS(QR-1.) LT.0.02) GO TO 200**
**IF(QR.LT.1.) GO TO 130**

**2ND STREAM LIMITING**

**J=1**
**GO TO 132**

**FIRST STREAM LIMITING**

**130 J=2**

**COMPUTE Q + NON LIMITING STREAM CONDITION**

**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**

**C**

**DATA**
****

****
ROUTINE TO CALC ENTHALPY CHANGE FOR STREAM J (HOJ-HIJ) OR COMPUTE OUTLET STREAM J CONDITION FOR GIVEN ENTHALPY

SAVE INPUT 1 IN INPUT 3
TRANSFER INPUT J TO INPUT 1

DO 166 II=1,3
   IF(II.EQ.II J) GO TO 166
   DO 162 K=1,7
      SI DUM(II J,K) = SIDUM(III,K)
   DO 164 I=1,NOCOMP
      XIN(I,II J) = XIN(I,III)
   II J=1

CONTINUE
SAVE OUTPUT (3-J) IN OUTPUT 3
DO 170 K=1,7
   SODUM(3,K) = SODUM(3-J,K)
   DO 174 I=1,NOCOMP
      XOUT(I,3) = XOUT(I,3-J)

NIN=NOUT=1
IF(IFL.EQ.2) GO TO 176
TIN(1)=IT
CALL ISOF(0.)
GO TO 178

HIN(1)=HH
CALL ADBF(0.)

RESTORE INPUT 1
DO 180 K=1,7
   SODUM(1:K) = SODUM(3,K)
   DO 182 I=1,NOCOMP
      XIN(I,1) = XIN(I,3)

RESET OUTPUTS 1 + 2 - OUTPUT 1 INTO J, THEN 3 INTO (3-J)
II J=J
I J=1

DO 186 K=1,7
   SODUM(II J,K) = SODUM(IJ,K)
   DO 188 I=1,NOCOMP
      XOUT(I,II J) = XOUT(I,I J)
   IF(II J.EQ.3-J) GO TO 190
   I J=3-J
   I J=3

GO TO 184

DH=HOUT(J)-HIN(J)
IF(IFL.EQ.1) GO TO 122

*****
** COMPUTE EXCHANGER AREA
** WORK IN DIRN OF INCR TEMP FOR STREAM 1
** DIVIDE INTO 10 SECTIONS + INTEGRATE NUMERICALLY WITH Q
** Assume T LINEAR WITH Q WITHIN EACH PHASE SEGMENT

COMPUTE HBP + HDP FOR EACH STREAM + SET UP BOUNDARY VALUES

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.*
      KVFK(KK)=VV
   VV=VFOUT(J)
   JVF(J)=1
   IF(KVF(1).EQ.KVF(2)) GO TO 204

200 VV=VFOUT(J)
202 VV=VFOUT(J)
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
CONTINUE

SET UP INPUTS/OUTPUTS TO GIVE INCR TEMP FOR STREAM 1

Q+ HEATING FIRST STREAM , - COOLING

JJ = J
IF(Q(J) .LT. 0. ) JJ = 3 - J
GO TO (206, 208) JJ

XTI(J) = TIN(J)
XTO(J) = TOUT(J)
IF(ISV(J) .EQ. 1) GO TO 210
XHI(J) = HIN(J)
XHO(J) = HOUT(J)
GO TO 210

XTI(J) = TOUT(J)
XTO(J) = TIN(J)
IF(ISV(J) .EQ. 1) GO TO 210
XHI(J) = HOUT(J)
XHO(J) = HIN(J)
GO TO 210

WRITE(*, 601) Q, XTI(J), XTO(J), XHI(J), XHO(J)

601 FORMAT(* Q10.0, 2F8.1, 5X, 2F8.1, 10X, 2F10.0, 5X, 2F10.0)

* INITIALIZE + INTEGRATE

DO = ABS(Q(J))/10.
DI = ABS(XTI(J) - XTI(J)) + QQ
RT0 = 1/XTI(J) - 1/XTI(J)
DENT = AREA = 0.

DO 240 K = 1, 10
QT = DO * FLOAT(K)
FRQ = QT/ABS(Q)

DO 230 J = 1, 2
IF(J .EQ. 2 AND IS2 .LT. 4) GO TO 226
HJ = XHI(J) + QT
IF(J .EQ. 15) 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

BP = BPIN(J)
DP = DPIN(J)
HB = HBP(J)
HD = HDP(J)

IF(BPIN(J) .GT. 260.) GO TO 214
BP = AMIN1(TIN(J), TOUT(J))
HB = AMIN1(HIN(J), HOUT(J))

C+ 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
VAPOR
AA=DP
BB=(HJ-HD)/(XHO(J)-HD)
CC=XTI(J)
GO TO 220

LIQUID
AA=XTI(J)
BB=(HB-HJ)/(HB-XHI(J))
CC=BP

EVALUATE U & S AT MIDPOINT
IF(JVF(J).*EQ.*0) GO TO 220
IVF=KVF(J)
GO TO 221

IF(J.*EQ.*LPS)

HHJ=HJ-0.5*DQ
IF(HHJ*GT.*HD) IVF=3
IF(HHJ*LT.*HB) IVF=1
GO TO (222,223,224)IVF

UU(J)=XU2
GO TO 225

UU(J)=XU1
GO TO 225

UU(J)=XU3
IF(PIN(J)*LT.*PLLOW) UU(J)=XU4

IF(K.*EQ.*10) GO TO 230
TT(J)=AA+BB*(CC-AA)
GO TO 230

TT(2)=XTI(2)+FRQ*DT2
UU(2)=U2
CONTINUE

ATC=

TEST=U*9*AP

IF(DTO*LT*TEST OR DTN*LT*TEST) WRITE(6,*730)DTO,DTN

730 FORMAT(* WARNING - BELOW APMPN - DTO,DTN*,2F8.1)

AREA=AREA+DQ/(U*XLMTD)

ENTROPY INCREASE
DENT=DENT+ABS(DQ*0.5*(RTO+RTN))
DTO=DTO+0.01
RTO=RTN

CONTINUE

CALCULATE COSTS

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

CYR=AMORT*CAP+COP

SET OUTPUT PARAMETERS

CALL ZERO(EQUIP,15)
EQUIP(2)=10P
EQUIP(7)=0
EQUIP(8) = AREA
EQUIP(9) = 1. E5 * DENT / ABS(Q)
IF(IS2.LT.3) EQUIP(10) = FLSRV
IF(IPS.GT.U) EQUIP(10) = QR*TMIN(IPS)
IF(IS2.EQ.3) EQUIP(11) = Tspec(2)
IF(IPS.GT.U) EQUIP(11) = Tspec(IPS)
EQUIP(12) = FM
EQUIP(13) = CAP
EQUIP(14) = COP
EQUIP(15) = CYR
RETURN
END
SUBROUTINE COMPIDIS

EQUIP TYPE 11

COMPRESSOR DESIGN ROUTINE
COMPUTES POWER REQUIREMENTS, CAPITAL+POWER COSTS FOR SINGLE
STAGE ISENTROPIC COMPRESSION

TOUT/TIN = (POUT/PIN)**((COEFF-1.)/COEFF)
Q ASSUMED ZERO, THEN W=-DELTA(H)

NOMENCLATURE -
PO*D - DISCHARGE PRESSURE
COEFF - ISENTROPIC TEMP COEFF
HRS - NO. OF OPERATING HOURS/YEAR
EFF - OVERALL MECHANICAL EFFICIENCY FACTOR
ACOMP, BCOMP - COST COEFFS. FOR COMPRESSOR
AMOT, BMOT - COST COEFFS. FOR COMPRESSOR MOTOR
CKWH - POWER COST/KWHR
CFCP - FACTOR TOT CAP INV/COMPR CAP COST

EQUIP OUTPUT VECTOR CODING -
1. - 2. EQUIP NO. + TYPE
3. - 6. INLET/OUTLET STREAM NOS
7. COMPRESSOR HP
8. - 9. INLET/OUTLET Pressures (PSIA)
13. CAPITAL COST ($) 
14. OPERATING (POWER) COST ($) /YR
15. TOTAL COST ($) /YR

C*** COMMON DECK
COMM/CONTL/NE,NIN,NOUT,NOCOMP
COMM/EQUIP/EQUIP(15)
COMM/SIN/BN(4),BN(4),LN(4),PIN(4),HN(4),VF(4),TM(4),
1 XIN(4),
COMM/OUT/BP(4),DP(4),TOUT(4),POUT(4),HOUT(4),VF(4),
1 TMOUT(4),XOUT(4)
COMM/PARAM/AMORT,HRS,SWAT,DTEMP,CWAT,TS,HV,CS,CKWH,APP,APR,
1 ARR,TRRR

C***

DIMENSION DUM(8)

DATA EFF,ACOMP,BCOMP,AMOT,BMOT,CFCP/0.9480,0.7634,1.12,5/
DATA CF1,CF2,NC3/1.31,0.15/

C

DO 1 I=1,NOCOMP
1 XOUT(I,1)=XIN(I,1)
TMOUT(I)=TMEN(I)
VFOUT(I)=1.
POUT(I)=PDIS
PR=POIS/PIN(1)
COMPUTE COEFF (= CF1 - CF2*FRACTION C3+)
FC3=0.
DO 2 I=NC3,NOCOMP
2 FC3=FC3+XIN(I,1)
FRC3=FC3/TMEN(I)
COEFF=CF1-CF2*FRC3
COMPUTE DISCHARGE TEMP
TOUT(I)=TMEN(I)*PR*((COEFF-1.)/COEFF)

COMPUTE OUTLET BUBBLE + DEW POINT TEMPS
CALL BUBTP(-1,BP(1),DUM)
CALL DEWTP(-1,DP(1),DUM)

COMPUTE ENTHALPY CHANGE + HP (ADJUST FOR EFFICIENCY)
CALL ENTH(-1,HOUT(1),DUM)
HP=(HOUT(I)-HIN(1))*3.93E-4/EFF
CPWR=0.746*HP*HRS*CKWH
CALCULATE CAPITAL + YEARLY COSTS

**CCOMP** = COST(ACOMP, BCOMP)
CMOT = COST(AMOT, BMOT)
CAP = CFCP * (CCOMP + CMOT)
CYR = CPWR + AMORT * CAP

SET OUTPUT PARAMETERS

CALL ZERO(EQUIP, 15)
EQUIP(2) = 1
EQUIP(7) = HP
EQUIP(8) = PIN(1)
EQUIP(9) = PDIS
EQUIP(13) = CAP
EQUIP(14) = CPWR
EQUIP(15) = CYR
RETURN
END
SUBROUTINE SPLIT(FR1)

***** EQUIP TYPE 20

SPLITS INPUT LINEARLY INTO 2 OUTPUTS

EQUIP OUTPUT VECTOR CODING -
1. - 2. EQUIP NO + TYPE
3. - 6. INLET/OUTLET STREAM NOS
7. - 8. 1ST, 2ND OUTLET STREAM FLOW FRACTIONS (*100)

***** COMMON DECK
COMMON/CONTLY, NIN, NOUT, NOCOMP
COMMON/EQUIP/EQUIP(15)
COMMON/SIN/BPIN(4), DPIN(4), TIN(4), PIN(4), HIN(4), VFIN(4), Tmin(4),
           XIN(8, 4)
COMMON/SOUT/BPOUT(4), DPOUT(4), TOUT(4), POUT(4), HOUT(4), Vfout(4),
           TMOUT(4), XOUT(8, 4)

*****
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
   XOUT(I, 2) = XX * FR2
1

** SET OUTPUT PARAMETERS

CALL ZERO(EQUIP, 15)
EQUIP(2) = 20.
EQUIP(7) = 100. * FR1
EQUIP(8) = 100. * FR2
RETURN
END
SUBROUTINE SPLINE(JCH)
SETS UP CUBIC SPLINE FOR ENERGY VALUE $/BTU (V) VS TEMP DEGREES (T)
X=TEMP, Y=VALUE, PM=MOMENTS
JCH=1 HOT (NH POINTS), 2 COLD (NC POINTS) - COMMON POINT NC+1
COMMON/SPLINE/NH,NC,X(10),Y(10),PM(10)
DIMENSION Q(10),U(10)

IF(JCH.EQ.2) GO TO 20
N1=NC+1
N2=NC+NH+1
GO TO 30
20 N1=1
N2=NC+1

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*C(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)=D(J)*PM(J+1)+U(J)
RETURN
END
SUBROUTINE SVALUE(JCH,TEX,TDSP,VALUE)

C***** EQUIP TYPE 30

COMPUTES INPUT STREAM ENERGY VALUE/YEAR
VALUE/BTU AS FUNCTION OF TEMP IS OBTAINED FROM INTERPOLATING SPLINE
ASSUMES 0 VS T LINEAR WITHIN EACH PHASE SEGMENT

JCH 1 HOT, 2 COLD
TEX EXIT TEMP SPEC
TDSP TEMP DISPLACEMENT

EQUIP OUTPUT VECTOR CODING -
1-2. EQUIP NO + TYPE
3-6. INLET/OUTLET STREAM NOS
7. HEAT AVAILABLE + REFR - WASTE HEAT (BTU/HR)
8-11. INLET/OUTLET TEMPS (DEG R)
14. OPERATING COST (CREDIT) ($/YR)
15. TOTAL COST ($/YR)

C***** COMMON DECK
COMMON/CONTL/NE,NIN,NOUT,NOCOMP
COMMON/EQUIP/EQUIP(15)
COMMON/IN/EPIN(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),XMOUT(8,4)
COMMON/PARAM/AMORT,HRS,SWAT,DTW,CWAT,TS,HWS,CS,CKWH,APPP,APRR,
1 ARR,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
CAC* HOT
XTOUT=TIN(1)
XHOUT=HIN(1)
TIN(1)=XTIN=TEX
CALL ISOF(0.,)
XHIN=HOUT(1)
GO TO 20
CAC* 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** INTEGRATE IN DIRECTION OF INCREASING TEMP
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.LE.(BP-U.1)) GO TO 95
TB=BP
VFIN(1)=0.
GO TO 90
C* 2 TWO-PHASE
30 IF(XTIN.GT.(DP+0.1)) GO TO 40
ASSIGN 40 TO IVAL
IF(XTOUT.LE.(DP-U.1)) GO TO 95
TB=DP
VFm(1)=1.
GO TO 90
C
C* 3 VAPOR
40 GO TO 95
C
50 TIN(1)=TSAV
VFm(1)=VSAV
C**
SET OUTPUT PARAMETERS
C
CALL ZERO(EQUIP,15)
EQUIP(2)=30.
EQUIP(7)=QT
EQUIP(8)=TIN(1)
EQUIP(9)=TEX
14 IS -VE (CREDIT)
VALUE=VALUE*HRS
EQUIP(15)=EQUIP(14)=-VALUE
RETURN
C
90 IFIN=0
TIN(1)=TB
CALL ENTH(1,TB,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)
COMPUTES ENERGY VALUE/BTU (YR) AT TEMP (XR) FROM CUBIC SPLINE
JCH=1 HOT, 2 COLD
COMMON/SPLINE/NH, NC, X(10), Y(10), PM(10)
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.

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 NPATHS 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 /PRLST/ 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.
### COMPONENT PHYSICAL PROPERTIES DATA

$COMP$

| NOCOMP=7,COMPNT=1,2,3,4,5,6,7,0 |

1. **HYDROGEN**
   - $1.8796E+02$  
   - $5.98860E+01$  
   - $1.04108E+00$  
   - $2.01600E+00$  
   - $9.56300E-07$

2. **METHANE**
   - $9.5000E-01$  
   - $6.95200E-00$  
   - $4.57600E-04$  
   - $9.56300E-07$

3. **ETHYLENE**
   - $4.2148E-02$  
   - $9.86060E+00$  
   - $2.80520E+00$  
   - $9.49000E-02$

4. **ETHANE**
   - $8.0000E+00$  
   - $8.80000E+00$  
   - $9.44000E-01$  
   - $1.99300E-05$

5. **PROPYLENE**
   - $8.0000E+00$  
   - $7.88000E+00$  
   - $8.20100E-02$  
   - $1.04900E-05$

6. **PROPANE**
   - $5.94292E-01$  
   - $3.72492E+00$  
   - $2.91229E-01$  
   - $3.00000E-06$

7. **N-BUTANE**
   - $1.04108E+00$  
   - $1.20496E-01$  
   - $2.01600E+00$  
   - $9.56300E-07$

8. **DETHANIZER**
   - $6.57180E+02$  
   - $2.89900E+00$  
   - $4.20708E+00$  
   - $1.45100E+01$

9. **C2 SPLITTER**
   - $9.69000E+00$  
   - $7.53000E-01$  
   - $5.69100E-02$  
   - $2.91000E-05$

10. **DEPROPANIZER**
    - $8.42550E+00$  
    - $5.39810E-01$  
    - $2.74296E-01$  
    - $5.22600E-01$

11. **C3 SPLITTER**
    - $6.17379E+02$  
    - $6.59460E+02$  
    - $4.09400E+01$  
    - $1.53800E-01$

12. **N-BUTANE**
    - $1.04108E+00$  
    - $1.20496E-01$  
    - $2.01600E+00$  
    - $9.56300E-07$

13. **DETHANIZER**
    - $6.57180E+02$  
    - $2.89900E+00$  
    - $4.20708E+00$  
    - $1.45100E+01$

14. **C2 SPLITTER**
    - $9.69000E+00$  
    - $7.53000E-01$  
    - $5.69100E-02$  
    - $2.91000E-05$

15. **DEPROPANIZER**
    - $8.42550E+00$  
    - $5.39810E-01$  
    - $2.74296E-01$  
    - $5.22600E-01$

16. **C3 SPLITTER**
    - $6.17379E+02$  
    - $6.59460E+02$  
    - $4.09400E+01$  
    - $1.53800E-01$

17. **N-BUTANE**
    - $1.04108E+00$  
    - $1.20496E-01$  
    - $2.01600E+00$  
    - $9.56300E-07$

18. **DETHANIZER**
    - $6.57180E+02$  
    - $2.89900E+00$  
    - $4.20708E+00$  
    - $1.45100E+01$

19. **C2 SPLITTER**
    - $9.69000E+00$  
    - $7.53000E-01$  
    - $5.69100E-02$  
    - $2.91000E-05$

20. **DEPROPANIZER**
    - $8.42550E+00$  
    - $5.39810E-01$  
    - $2.74296E-01$  
    - $5.22600E-01$

21. **C3 SPLITTER**
    - $6.17379E+02$  
    - $6.59460E+02$  
    - $4.09400E+01$  
    - $1.53800E-01$

### COLUMN SYSTEM DATA

$HP$ COLUMN SYSTEM

$SPARLST$

|$AMOR=0.3,HRS=8000.$
|$TWAT=532,DTW=10.$
|$CWAT=3,6E-5,TS=825.$
|$HVS=1.00E3,CS=1.80E-2,CKWH=0.007$

$APP=15.$

$APRR=10.$

$TRR=310.$

$END$

$C** COLUMN NO.$

1. **1 DEMETHANIZER**
2. **2 DEETHANIZER**
3. **3 C2 SPLITTER**
4. **4 DEPROPANIZER**
5. **5 C3 SPLITTER**

$KXMLST$

|$NIS=11,NKPM=5.$
|$KPM1=1,100,-2,-3,0.$
|$KPM2=2,100,3,-4,-5,0.$
|$KPM3=3,100,4,-6,-7,0.$
|$KPM4=4,100,5,-8,-9,0.$
|$KPM5=5,100,8,-10,11,0.$

$END$

$SPMLST$

|$SMPA1=1,1,1,1,1,565,0,400,2,0,0.$
|$SMPA2=2,-1,1,1215,0,-1,2,0,0.$
|$SMPA7=-7,-1,1,115,2,-1,2,0,0.$
|$SMPB1=2,0,520,115,0,1,1,255,495,315,210,135,45,45,0,0.$

$END$

$EMILST$

|$EM1=1,100,1,1,1,0,65,0,01,0,01,0,043,2,3,5,65,24,3,0,0.$
|$EM2=2,100,0,1,2,0,385,0,01,0,025,0,565,4,5,465,24,3,0,0.$
|$EM3=3,100,0,1,2,0,96,0,01,0,017,0,93,3,4,215,18,3,0,0.$
|$EM4=4,100,0,1,2,0,25,0,035,0,04,0,835,6,7,200,24,3,0,0.$
|$EM5=5,100,0,1,2,0,90,0,08,0,075,0,76,5,6,115,18,3,0,0.$

$END$
<table>
<thead>
<tr>
<th></th>
<th>100</th>
<th>0</th>
<th>1.00</th>
<th>300.5</th>
<th>1.28</th>
<th>9</th>
<th>11</th>
<th>2.27</th>
<th>0.78</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.72</td>
<td>13120</td>
<td>99361</td>
<td>0</td>
<td>29808</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>18694</td>
<td>87835</td>
<td>0</td>
<td>26351</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.00</td>
<td>19039</td>
<td>145221</td>
<td>0</td>
<td>43566</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1.00</td>
<td>2602</td>
<td>23424</td>
<td>0</td>
<td>7027</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1.00</td>
<td>20898</td>
<td>226055</td>
<td>0</td>
<td>67817</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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, PARLIST.

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.
**HP PLANT - PROpane SERIES PROCESSED**

```plaintext
$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
```

**C:. HOT STREAM DATA -**

<table>
<thead>
<tr>
<th>HOT CONTROL VECTORS</th>
<th>THEN ALTERNATE PROPERTIES + FLOWS VECTORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1    0   0</td>
<td>0   0 555.0</td>
</tr>
<tr>
<td>2    0   0</td>
<td>1   0 565.0 400.0</td>
</tr>
<tr>
<td>3    0   0</td>
<td>0   0 301.5</td>
</tr>
<tr>
<td>4    0   0</td>
<td>0   0 481.9</td>
</tr>
<tr>
<td>5    0   0</td>
<td>0   0 416.0</td>
</tr>
<tr>
<td>6    0   0</td>
<td>0   0 512.0</td>
</tr>
<tr>
<td>7    0   0</td>
<td>0   0 508.0</td>
</tr>
</tbody>
</table>

| 555.2  | 555.2  | 614.4 |
| 0.00   | 0.00   | 0.00  |
| 240.0  | 449.1  | 520.0 |
| 255.0  | 495.0  | 315.0 |
| 240.0  | 348.5  | 348.5 |
| 255.0  | 1041.2 | 122.2 |
| 481.9  | 490.9  | 490.9 |
| 0.00   | 14.05  | 628.7 |
| 416.0  | 422.1  | 422.1 |
| 0.00   | 23.90  | 1056.26|
| 552.0  | 559.7  | 559.7 |
| 0.00   | 0.00   | 0.00  |
| 508.2  | 512.5  | 512.5 |
| 0.00   | 0.00   | 0.00  |

| 115.0 | 174.4 | 15604605 | 1.000 | 2.368.7 |
| 0.00 | 0.00 | 22214.40 | 0.00 |
| 215.0 | 465.0 | 2140741 | 1.000 | 45.00 |
| 200.0 | 9.01 | 196.85 | 1.000 | 66.71 |
| 115.0 | 3802021 | 1.000 | 987.8 |
| 40.74 | 868.85 | 78.18 |

**C:. COLD STREAM DATA**

<table>
<thead>
<tr>
<th>COLD CONTROL VECTORS</th>
<th>THEN ALTERNATE PROPERTIES + FLOWS VECTORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1    0   0</td>
<td>0   0 577.8</td>
</tr>
<tr>
<td>2    0   0</td>
<td>0   0 648.6</td>
</tr>
<tr>
<td>3    0   0</td>
<td>0   0 653.8</td>
</tr>
<tr>
<td>4    0   0</td>
<td>0   0 547.6</td>
</tr>
<tr>
<td>5    0   0</td>
<td>0   0 565.0</td>
</tr>
<tr>
<td>6    0   0</td>
<td>0   0 925856</td>
</tr>
<tr>
<td>7    0   0</td>
<td>0   0 925856</td>
</tr>
<tr>
<td>240.0</td>
<td>300.1</td>
</tr>
<tr>
<td>255.0</td>
<td>488.16</td>
</tr>
<tr>
<td>530.0</td>
<td>577.8</td>
</tr>
<tr>
<td>0.00</td>
<td>9.36</td>
</tr>
<tr>
<td>634.1</td>
<td>648.6</td>
</tr>
<tr>
<td>0.00</td>
<td>4.61</td>
</tr>
<tr>
<td>461.3</td>
<td>466.6</td>
</tr>
<tr>
<td>0.00</td>
<td>3.73</td>
</tr>
<tr>
<td>461.3</td>
<td>466.6</td>
</tr>
<tr>
<td>0.00</td>
<td>22.44</td>
</tr>
<tr>
<td>644.5</td>
<td>653.8</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>533.7</td>
<td>547.6</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

| 565.0 | 565.0 | 925856 | 1.000 | 751.6 |
| 0.00  | 0.00  | 1023.9 |
| 465.0 | 465.0 | 143780 | 0.000 | 1011.0 |
| 577.15 | 577.15 | 199.15 |
| 465.0 | 465.0 | 581610 | 0.000 | 203.8 |
| 533.7 | 533.7 | 72.28  | 741.46 | 145.49 |

**C:. REFRIGERATION LEVELS -**

<table>
<thead>
<tr>
<th>REFRIGERATION LEVELS</th>
</tr>
</thead>
<tbody>
<tr>
<td>296.12</td>
</tr>
<tr>
<td>345.12</td>
</tr>
<tr>
<td>385.12</td>
</tr>
<tr>
<td>425.12</td>
</tr>
<tr>
<td>470.12</td>
</tr>
</tbody>
</table>
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 \( \delta \) 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  BRNBD SECTION INPUT ***

HP PLANT - PROPANE SERIES PROCESSED

$PARLST
AMOT=0.3, HRS=8000,
TWAI=535., DTW=6E-5, TS=825., HVS=1.00E3, CS=1.80E-2, CKWH=0.007,
APPP=1., APRR=1., ARRR=5., TRR=310., DTF(2)=0.,
$END

C. BRANCH + BOUND PARAMETER CARD

10 3 4 2

C. HOT STREAM MOLE FRACTIONS

<table>
<thead>
<tr>
<th>Branch</th>
<th>Mol Fraction 1</th>
<th>Mol Fraction 2</th>
<th>Mol Fraction 3</th>
<th>Mol Fraction 4</th>
<th>Mol Fraction 5</th>
<th>Mol Fraction 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>0.1700</td>
<td>0.3300</td>
<td>0.2100</td>
<td>0.1400</td>
<td>0.0900</td>
<td>0.0300</td>
</tr>
<tr>
<td>3</td>
<td>0.4818</td>
<td>0.7332</td>
<td>0.0861</td>
<td>0.0006</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>4</td>
<td>0.0000</td>
<td>0.0130</td>
<td>0.5824</td>
<td>0.3875</td>
<td>0.0149</td>
<td>0.0000</td>
</tr>
<tr>
<td>5</td>
<td>0.0000</td>
<td>0.0219</td>
<td>0.9682</td>
<td>0.0099</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>6</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0319</td>
<td>0.6973</td>
<td>0.2363</td>
</tr>
<tr>
<td>7</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0412</td>
<td>0.8796</td>
<td>0.0791</td>
</tr>
</tbody>
</table>

C. HOT STREAM CONTROL + PROPERTIES VECTORS

<table>
<thead>
<tr>
<th>Branch</th>
<th>Control 1</th>
<th>Control 2</th>
<th>Control 3</th>
<th>Properties 1</th>
<th>Properties 2</th>
<th>Properties 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>555.2</td>
<td>555.2</td>
<td>614.4</td>
<td>174.4</td>
<td>15604605</td>
<td>0.0000</td>
</tr>
<tr>
<td>2</td>
<td>174.4</td>
<td>15604605</td>
<td>0.0000</td>
<td>555.2</td>
<td>2214.4</td>
<td>1.0000</td>
</tr>
<tr>
<td>3</td>
<td>552.0</td>
<td>555.2</td>
<td>512.5</td>
<td>555.2</td>
<td>2214.4</td>
<td>1.0000</td>
</tr>
<tr>
<td>4</td>
<td>555.2</td>
<td>2214.4</td>
<td>1.0000</td>
<td>555.2</td>
<td>2214.4</td>
<td>1.0000</td>
</tr>
<tr>
<td>5</td>
<td>555.2</td>
<td>2214.4</td>
<td>1.0000</td>
<td>555.2</td>
<td>2214.4</td>
<td>1.0000</td>
</tr>
<tr>
<td>6</td>
<td>555.2</td>
<td>2214.4</td>
<td>1.0000</td>
<td>555.2</td>
<td>2214.4</td>
<td>1.0000</td>
</tr>
<tr>
<td>7</td>
<td>555.2</td>
<td>2214.4</td>
<td>1.0000</td>
<td>555.2</td>
<td>2214.4</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

C. COLD STREAM DATA

<table>
<thead>
<tr>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
<th>Value 4</th>
<th>Value 5</th>
<th>Value 6</th>
<th>Value 7</th>
<th>Value 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.0091</td>
<td>0.0000</td>
<td>0.046</td>
<td>0.0255</td>
<td>0.5709</td>
<td>0.1970</td>
<td>0.2020</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>17</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>20</td>
<td>21</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>22</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>23</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>25</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>26</td>
<td>37</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

**COLD STREAMS**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>3</th>
<th>1</th>
<th>3</th>
<th>6</th>
<th>1</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>23</td>
<td>32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>1</td>
<td>115.0</td>
<td>254.9</td>
<td>0</td>
<td>0</td>
<td>8</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>684.5</td>
<td>13.7</td>
<td>202</td>
<td>7654.1</td>
<td>9</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>254.9</td>
<td>565.0</td>
<td>0.0</td>
<td>0.0</td>
<td>10</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>828.1</td>
<td>17.8</td>
<td>11508.4</td>
<td>11</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>565.0</td>
<td>215.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-8</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>6</td>
<td>215.0</td>
<td>115.0</td>
<td>0.0</td>
<td>0.0</td>
<td>-9</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>98.4</td>
<td>59.4</td>
<td>246.2</td>
<td>0</td>
<td>0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>146.9</td>
<td>103.2</td>
<td>47.0</td>
<td>141.7</td>
<td>249.0</td>
<td>425.0</td>
<td>8764.9</td>
</tr>
<tr>
<td></td>
<td>-3</td>
<td>-5</td>
<td>-6</td>
<td>-7</td>
<td>-8</td>
<td>-9</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>34.7</td>
<td>94.3</td>
<td>32.8</td>
<td>63.8</td>
<td>0.0</td>
<td>0.0</td>
<td>6.7</td>
</tr>
<tr>
<td></td>
<td>201</td>
<td>201</td>
<td>201</td>
<td>201</td>
<td>0.0</td>
<td>0.0</td>
<td>95320.3</td>
</tr>
<tr>
<td></td>
<td>249.6</td>
<td>344.9</td>
<td>76.5</td>
<td>373.1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>6208896</td>
<td>1377391</td>
<td>6715581</td>
<td>6432222</td>
<td>0.0</td>
<td>0.0</td>
<td>142625</td>
</tr>
<tr>
<td></td>
<td>5297</td>
<td>3305</td>
<td>6408</td>
<td>64331</td>
<td>0.0</td>
<td>0.0</td>
<td>27452</td>
</tr>
</tbody>
</table>
|   | 35946 | 51260 | 53725 | 19299 | 0.0 | 0.0 | 70240 | 19299 | 0.0 | 0.0 | 53608 | 0.0 | 11171 | 21816 | 45643 | 0 | 0 | 0 | 0 | 2726953 | 130811 | 0 | 2726953 | 3921 | 22992
33  106.2  63.2  201  277.9  0  0  5002857  1.0  5390  40023  41640
34  6958.8  12  202  0  0  -12725440  1.0  118584  20361  55936
35  726.0  35.4  -5  0.0  17  0  -6516545  1.0  69423  10426  31253
36  4509.1  12  -7  18  -14  -6089290  1.0  83806  0  25142
37  0.0  14  203  508.2  0  0  0.0  0  0  0
38  0.0  16  203  0  0  0.0  0  0  0
39  224.0  16  -9  20  0  -1125983  1.0  8434  0  2530
40  13.5  61.6  34.8  0  0  626291  1.0  1564  5010  5480
41  3563.5  17  202  0  0  -6516545  1.0  69423  10426  31253
42  4509.1  17  -7  21  -15  -6089290  1.0  83806  0  25142
43  3628.9  18  202  0  0  -6636151  1.0  70440  10618  31750
44  726.0  18  -5  22  0  -6208896  1.0  19442  0  5833
45  0.0  17  203  0  0  0.0  0  0  0
46  285.8  19  -9  23  -16  -500273  1.0  9761  0  2928
47  0.0  20  203  24  0  0.0  0  0  0
48  13.5  61.6  34.8  0  0  626291  1.0  1564  5010  5480
49  -1.0  -16  0  0.0  0  0.0  0  0  0
50  233.6  21  202  0  0  -427255  1.0  8650  684  3279
51  233.6  22  202  0  0  -427255  1.0  8650  684  3279
52  0.0  23  203  0  0  0.0  0  0  0
53  0.0  24  203  0  0  0.0  0  0  0

C. REFRIGERATION LEVELS + COSTS -

296:  0.000138
345:  0.000120
385:  0.000095
425:  0.000043
470:  0.000025
### HP (SORTED) STREAM PROCESSING PATH ARRAY ###

#### * DENOTES OPTIMUM PROCESSING SEQUENCE

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
<td>18</td>
</tr>
<tr>
<td>19</td>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
</tr>
<tr>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
<td>30</td>
</tr>
<tr>
<td>31</td>
<td>32</td>
<td>33</td>
<td>34</td>
<td>35</td>
<td>36</td>
</tr>
<tr>
<td>37</td>
<td>38</td>
<td>39</td>
<td>40</td>
<td>41</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>44</td>
<td>45</td>
<td>46</td>
<td>47</td>
<td>48</td>
</tr>
<tr>
<td>49</td>
<td>50</td>
<td>51</td>
<td>52</td>
<td>53</td>
<td>54</td>
</tr>
<tr>
<td>55</td>
<td>56</td>
<td>57</td>
<td>58</td>
<td>59</td>
<td>60</td>
</tr>
<tr>
<td>61</td>
<td>62</td>
<td>63</td>
<td>64</td>
<td>65</td>
<td>66</td>
</tr>
<tr>
<td>67</td>
<td>68</td>
<td>69</td>
<td>70</td>
<td>71</td>
<td>72</td>
</tr>
<tr>
<td>73</td>
<td>74</td>
<td>75</td>
<td>76</td>
<td>77</td>
<td>78</td>
</tr>
<tr>
<td>79</td>
<td>80</td>
<td>81</td>
<td>82</td>
<td>83</td>
<td>84</td>
</tr>
<tr>
<td>85</td>
<td>86</td>
<td>87</td>
<td>88</td>
<td>89</td>
<td>90</td>
</tr>
<tr>
<td>91</td>
<td>92</td>
<td>93</td>
<td>94</td>
<td>95</td>
<td>96</td>
</tr>
<tr>
<td>97</td>
<td>98</td>
<td>99</td>
<td>100</td>
<td>101</td>
<td>102</td>
</tr>
<tr>
<td>103</td>
<td>104</td>
<td>105</td>
<td>106</td>
<td>107</td>
<td>108</td>
</tr>
<tr>
<td>109</td>
<td>110</td>
<td>111</td>
<td>112</td>
<td>113</td>
<td>114</td>
</tr>
<tr>
<td>115</td>
<td>116</td>
<td>117</td>
<td>118</td>
<td>119</td>
<td>120</td>
</tr>
<tr>
<td>121</td>
<td>122</td>
<td>123</td>
<td>124</td>
<td>125</td>
<td>126</td>
</tr>
<tr>
<td>127</td>
<td>128</td>
<td>129</td>
<td>130</td>
<td>131</td>
<td>132</td>
</tr>
<tr>
<td>133</td>
<td>134</td>
<td>135</td>
<td>136</td>
<td>137</td>
<td>138</td>
</tr>
<tr>
<td>139</td>
<td>140</td>
<td>141</td>
<td>142</td>
<td>143</td>
<td>144</td>
</tr>
<tr>
<td>145</td>
<td>146</td>
<td>147</td>
<td>148</td>
<td>149</td>
<td>150</td>
</tr>
<tr>
<td>151</td>
<td>152</td>
<td>153</td>
<td>154</td>
<td>155</td>
<td>156</td>
</tr>
<tr>
<td>157</td>
<td>158</td>
<td>159</td>
<td>160</td>
<td>161</td>
<td>162</td>
</tr>
<tr>
<td>163</td>
<td>164</td>
<td>165</td>
<td>166</td>
<td>167</td>
<td>168</td>
</tr>
<tr>
<td>169</td>
<td>170</td>
<td>171</td>
<td>172</td>
<td>173</td>
<td>174</td>
</tr>
<tr>
<td>175</td>
<td>176</td>
<td>177</td>
<td>178</td>
<td>179</td>
<td>180</td>
</tr>
<tr>
<td>181</td>
<td>182</td>
<td>183</td>
<td>184</td>
<td>185</td>
<td>186</td>
</tr>
<tr>
<td>187</td>
<td>188</td>
<td>189</td>
<td>190</td>
<td>191</td>
<td>192</td>
</tr>
<tr>
<td>193</td>
<td>194</td>
<td>195</td>
<td>196</td>
<td>197</td>
<td>198</td>
</tr>
<tr>
<td>199</td>
<td>200</td>
<td>201</td>
<td>202</td>
<td>203</td>
<td>204</td>
</tr>
<tr>
<td>205</td>
<td>206</td>
<td>207</td>
<td>208</td>
<td>209</td>
<td>210</td>
</tr>
<tr>
<td>211</td>
<td>212</td>
<td>213</td>
<td>214</td>
<td>215</td>
<td>216</td>
</tr>
<tr>
<td>217</td>
<td>218</td>
<td>219</td>
<td>220</td>
<td>221</td>
<td>222</td>
</tr>
<tr>
<td>223</td>
<td>224</td>
<td>225</td>
<td>226</td>
<td>227</td>
<td>228</td>
</tr>
<tr>
<td>229</td>
<td>230</td>
<td>231</td>
<td>232</td>
<td>233</td>
<td>234</td>
</tr>
<tr>
<td>235</td>
<td>236</td>
<td>237</td>
<td>238</td>
<td>239</td>
<td>240</td>
</tr>
</tbody>
</table>

**Note:** The numbers in the table represent the optimum processing sequence for various paths in the HP (Sorted) stream processing path array. Each row corresponds to a different path, with the numbers indicating the order of processing. The asterisks (*) denote the optimal sequence.
### C*** HP OPTIMAL PLANT SUMMARY ***

**OPTIMAL PLANT NO., COST**

<table>
<thead>
<tr>
<th></th>
<th>51</th>
<th>12</th>
<th>32</th>
<th>11</th>
<th>33</th>
<th>11</th>
<th></th>
<th>1030575</th>
</tr>
</thead>
</table>

**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**

<table>
<thead>
<tr>
<th>REFD, REFS, PRSV ARRAYS</th>
<th>296</th>
<th>390</th>
<th>435</th>
<th>472</th>
<th>485</th>
<th>249</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2394842</td>
<td>1521120</td>
<td>1854210</td>
<td>3926780</td>
<td>1744371</td>
<td>-8</td>
</tr>
</tbody>
</table>

**TOTAL PLANT COST SUMMARY**

<table>
<thead>
<tr>
<th>TOTAL</th>
<th>1030575</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS NETWORK (EXCL. REFR. DEMANDS + SALES)</td>
<td>558506</td>
</tr>
<tr>
<td>PRE-PROCESSING (PRESSURE SPECS)</td>
<td>218498</td>
</tr>
<tr>
<td>POST-PROCESSING (TEMPERATURE SPECS)</td>
<td>340008</td>
</tr>
<tr>
<td>REFRIGERATION DEMANDS</td>
<td>566180</td>
</tr>
<tr>
<td>STREAM SALES</td>
<td>-94111</td>
</tr>
</tbody>
</table>


C** HOT STREAM DATA **

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td></td>
</tr>
</tbody>
</table>

C** COLD STREAM DATA **

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3393</td>
<td>6495</td>
<td>0104</td>
<td>0008</td>
<td>00000</td>
<td>00000</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0091</td>
<td>0.4105</td>
<td>0.2796</td>
<td>0.1804</td>
<td>0.0601</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0176</td>
<td>0.9454</td>
<td>0.0807</td>
<td>0.4736</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0582</td>
<td>0.0411</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

C** OPTIMAL PLANT DETAILS **

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C. EQUIPMENT DATA</td>
<td>0.0</td>
<td>700</td>
<td>233843</td>
<td>29240</td>
<td>99393</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----</td>
<td>-----</td>
<td>---------</td>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>1 115.0 254.9 0.0 0.0 8.0 0.0</td>
<td>1.0</td>
<td>18549</td>
<td>2204</td>
<td>7769</td>
<td></td>
</tr>
<tr>
<td>2 684.5 13.7 202 7654.1 9.0 0.0</td>
<td>1.0</td>
<td>717</td>
<td>238590</td>
<td>29964</td>
<td>101541</td>
</tr>
<tr>
<td>3 254.9 9.0 0.0 0.0 10.0 0.0</td>
<td>1.0</td>
<td>2071520</td>
<td>3314</td>
<td>9795</td>
<td></td>
</tr>
<tr>
<td>4 828.1 17.8 102 11508.4 11.0 0.0</td>
<td>1.0</td>
<td>100</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>5 565.0 215.0 0.0 0.0 -8.0 0.0</td>
<td>1.0</td>
<td>16.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>6 215.0 115.0 0.0 0.0 -9.0 0.0</td>
<td>4493258</td>
<td>6547</td>
<td>35946</td>
<td>37910</td>
<td></td>
</tr>
<tr>
<td>7 146.9 34.7 -3.0 202 249.6 0.0 0.0</td>
<td>1.0</td>
<td>1377391</td>
<td>3305</td>
<td>11019</td>
<td>12011</td>
</tr>
<tr>
<td>8 47.0 32.8 -6.0 202 76.5 0.0 0.0</td>
<td>1.0</td>
<td>1555873</td>
<td>0.0</td>
<td>-94111</td>
<td>-94111</td>
</tr>
<tr>
<td>9 -1.0 0.0 0.0 0.0 0.0 0.0</td>
<td>-4432222</td>
<td>1.0</td>
<td>64331</td>
<td>0.0</td>
<td>19299</td>
</tr>
<tr>
<td>10 3239.9 6.6 12.0 0.0 0.0 0.0</td>
<td>-2394842</td>
<td>3.5</td>
<td>82186</td>
<td>264391</td>
<td>289046</td>
</tr>
<tr>
<td>11 919.0 30.3 203 301.5 0.0 296.0</td>
<td>-3926780</td>
<td>1.0</td>
<td>43146</td>
<td>78540</td>
<td>91484</td>
</tr>
<tr>
<td>12 513.0 8.0 4.0 203 481.9 0.0 470.0</td>
<td>0.0</td>
<td>292</td>
<td>114499</td>
<td>12194</td>
<td>46543</td>
</tr>
<tr>
<td>13 215.0 5.0 0.0 13.0 0.0 0.0</td>
<td>-4865113</td>
<td>1.0</td>
<td>63573</td>
<td>0.0</td>
<td>19072</td>
</tr>
<tr>
<td>14 3192.2 7.3 -5.0 0.0 0.0 0.0</td>
<td>-1703878</td>
<td>1.0</td>
<td>22295</td>
<td>2726</td>
<td>9415</td>
</tr>
<tr>
<td>15 861.5 5.3 0.0 202 9466.0 0.0 0.0</td>
<td>1.0</td>
<td>344</td>
<td>130811</td>
<td>14365</td>
<td>53608</td>
</tr>
<tr>
<td>16 27 115.0 222.4 7.0 0.0 15.0 0.0</td>
<td>0.0</td>
<td>6628207</td>
<td>1.0</td>
<td>69965</td>
<td>0.0</td>
</tr>
<tr>
<td>17 3598.3 15.0 -7.0 0.0 0.0 0.0</td>
<td>-1744371</td>
<td>1.0</td>
<td>10350</td>
<td>34800</td>
<td>45150</td>
</tr>
<tr>
<td>18 296.0 11.0 203 495.0 16.0 0.0</td>
<td>1.0</td>
<td>1317738</td>
<td>1.0</td>
<td>2097</td>
<td>10542</td>
</tr>
<tr>
<td>19 224.0 20.5 -11.0 201 73.2 0.0 0.0</td>
<td>1.0</td>
<td>-12725440</td>
<td>1.0</td>
<td>118584</td>
<td>20361</td>
</tr>
<tr>
<td>20 6958.8 12.0 202 70696.9 0.0 0.0</td>
<td>1.0</td>
<td>-1125983</td>
<td>1.0</td>
<td>8434</td>
<td>0.0</td>
</tr>
<tr>
<td>21 39 5.1 16.0 -9.0 0.0 20.0 0.0</td>
<td>-1847872</td>
<td>1.0</td>
<td>16969</td>
<td>64306</td>
<td>69397</td>
</tr>
<tr>
<td>22 22 16.0 -9.0 0.0 0.0 0.0</td>
<td>1.0</td>
<td>-1515039</td>
<td>2.0</td>
<td>24946</td>
<td>115143</td>
</tr>
</tbody>
</table>
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,
THAT=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=31V., DTF(2)=0.,

$END

C.. PREVIOUS REFRIGERATION LEVELS + COSTS -

<table>
<thead>
<tr>
<th>Level</th>
<th>Previous Level</th>
<th>Previous Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>296</td>
<td>0.00000133</td>
<td></td>
</tr>
<tr>
<td>345</td>
<td>0.00000114</td>
<td></td>
</tr>
<tr>
<td>385</td>
<td>0.0000096</td>
<td></td>
</tr>
<tr>
<td>425</td>
<td>0.0000044</td>
<td></td>
</tr>
<tr>
<td>470</td>
<td>0.0000024</td>
<td></td>
</tr>
</tbody>
</table>

C.. NEW REFRIGERATION DEMANDS -

<table>
<thead>
<tr>
<th>Level</th>
<th>New Demand</th>
</tr>
</thead>
<tbody>
<tr>
<td>296</td>
<td>2400000</td>
</tr>
<tr>
<td>345</td>
<td>0</td>
</tr>
<tr>
<td>385</td>
<td>1520000</td>
</tr>
<tr>
<td>425</td>
<td>1850000</td>
</tr>
<tr>
<td>470</td>
<td>5670000</td>
</tr>
</tbody>
</table>

C.. PURCHASED STREAM INFORMATION -

<table>
<thead>
<tr>
<th>Stream</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>240.0</td>
</tr>
<tr>
<td>2</td>
<td>240.0</td>
</tr>
<tr>
<td>248.6</td>
<td></td>
</tr>
<tr>
<td>215.0</td>
<td>925856</td>
</tr>
<tr>
<td>1.00</td>
<td>751.6</td>
</tr>
<tr>
<td>0.3393</td>
<td>0.6495</td>
</tr>
<tr>
<td>0.0104</td>
<td>0.0008</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
### Ethylene Section Stream Vectors

<table>
<thead>
<tr>
<th></th>
<th>240.0</th>
<th>240.0</th>
<th>248.6</th>
<th>215.0</th>
<th>925856</th>
<th>1.000</th>
<th>751.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>435.0</td>
<td>435.0</td>
<td>435.0</td>
<td>263.9</td>
<td>-2839</td>
<td>0.000</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>435.0</td>
<td>435.0</td>
<td>371.6</td>
<td>263.9</td>
<td>-2882024</td>
<td>0.000</td>
<td>694.7</td>
</tr>
<tr>
<td>4</td>
<td>240.0</td>
<td>240.0</td>
<td>429.0</td>
<td>215.0</td>
<td>1835877</td>
<td>1.000</td>
<td>751.6</td>
</tr>
<tr>
<td>5</td>
<td>296.1</td>
<td>296.1</td>
<td>296.1</td>
<td>11.5</td>
<td>461059</td>
<td>1.000</td>
<td>419.2</td>
</tr>
<tr>
<td>6</td>
<td>435.0</td>
<td>435.0</td>
<td>347.4</td>
<td>263.9</td>
<td>-1938941</td>
<td>0.000</td>
<td>419.2</td>
</tr>
<tr>
<td>7</td>
<td>296.1</td>
<td>296.1</td>
<td>361.6</td>
<td>11.5</td>
<td>661070</td>
<td>1.000</td>
<td>419.2</td>
</tr>
<tr>
<td>8</td>
<td>385.2</td>
<td>385.2</td>
<td>371.6</td>
<td>106.9</td>
<td>385959</td>
<td>1.000</td>
<td>273.9</td>
</tr>
<tr>
<td>9</td>
<td>345.1</td>
<td>345.1</td>
<td>498.2</td>
<td>44.5</td>
<td>10029285</td>
<td>1.000</td>
<td>419.2</td>
</tr>
<tr>
<td>10</td>
<td>385.2</td>
<td>385.2</td>
<td>612.8</td>
<td>106.9</td>
<td>1643295</td>
<td>1.000</td>
<td>419.2</td>
</tr>
<tr>
<td>11</td>
<td>385.2</td>
<td>385.2</td>
<td>545.0</td>
<td>106.9</td>
<td>1319459</td>
<td>1.000</td>
<td>419.2</td>
</tr>
<tr>
<td>12</td>
<td>385.2</td>
<td>385.2</td>
<td>480.2</td>
<td>106.9</td>
<td>1705418</td>
<td>1.000</td>
<td>692.5</td>
</tr>
<tr>
<td>13</td>
<td>435.2</td>
<td>435.2</td>
<td>594.8</td>
<td>263.9</td>
<td>2446804</td>
<td>1.000</td>
<td>692.5</td>
</tr>
<tr>
<td>14</td>
<td>435.2</td>
<td>435.2</td>
<td>545.0</td>
<td>263.9</td>
<td>2032981</td>
<td>1.000</td>
<td>692.5</td>
</tr>
</tbody>
</table>

### Ethylene Section Equipment Vectors

<table>
<thead>
<tr>
<th></th>
<th>505.2</th>
<th>55.7</th>
<th>1.00</th>
<th>3.50</th>
<th>59011</th>
<th>0.000</th>
<th>15273</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10</td>
<td>6</td>
<td>0.0</td>
<td>0.0</td>
<td>16</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>3</td>
<td>297.0</td>
<td>26.8</td>
<td>5.0</td>
<td>6.7</td>
<td>-200011</td>
<td>34962</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>106.9</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>213</td>
<td>88638</td>
<td>8891</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>44.5</td>
<td>0.0</td>
<td>10.0</td>
<td>216</td>
<td>89731</td>
<td>9027</td>
</tr>
<tr>
<td>7</td>
<td>176.9</td>
<td>12.2</td>
<td>1799.1</td>
<td>11.0</td>
<td>323837</td>
<td>7321</td>
<td>518</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
<td>106.9</td>
<td>11.0</td>
<td>12.0</td>
<td>100</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>106.9</td>
<td>263.9</td>
<td>12.0</td>
<td>13.0</td>
<td>324</td>
<td>124553</td>
<td>13525</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>13</td>
<td>202</td>
<td>14.0</td>
<td>-413823</td>
<td>9545</td>
<td>662</td>
</tr>
<tr>
<td>11</td>
<td>2666.2</td>
<td>11.2</td>
<td>14</td>
<td>203</td>
<td>425.0</td>
<td>6012117</td>
<td>55045</td>
</tr>
</tbody>
</table>

### Propane Section Stream Vectors

<table>
<thead>
<tr>
<th></th>
<th>240.0</th>
<th>240.0</th>
<th>425.0</th>
<th>215.0</th>
<th>1835877</th>
<th>1.000</th>
<th>751.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>545.0</td>
<td>545.0</td>
<td>545.0</td>
<td>171.4</td>
<td>-960</td>
<td>0.000</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>545.0</td>
<td>545.0</td>
<td>535.3</td>
<td>174.4</td>
<td>-2765597</td>
<td>0.000</td>
<td>22078</td>
</tr>
<tr>
<td>4</td>
<td>240.0</td>
<td>240.0</td>
<td>535.0</td>
<td>215.0</td>
<td>2481729</td>
<td>1.000</td>
<td>751.6</td>
</tr>
<tr>
<td>5</td>
<td>425.2</td>
<td>425.2</td>
<td>425.2</td>
<td>18.2</td>
<td>4372435</td>
<td>1.000</td>
<td>1189.2</td>
</tr>
<tr>
<td>6</td>
<td>470.0</td>
<td>470.0</td>
<td>470.0</td>
<td>45.1</td>
<td>4385829</td>
<td>1.000</td>
<td>1025.2</td>
</tr>
<tr>
<td>7</td>
<td>470.0</td>
<td>470.0</td>
<td>497.3</td>
<td>45.1</td>
<td>5643455</td>
<td>1.000</td>
<td>1189.2</td>
</tr>
<tr>
<td>8</td>
<td>470.0</td>
<td>470.0</td>
<td>484.8</td>
<td>45.1</td>
<td>10029285</td>
<td>1.000</td>
<td>2214.4</td>
</tr>
<tr>
<td>9</td>
<td>555.2</td>
<td>555.2</td>
<td>598.2</td>
<td>174.4</td>
<td>13987886</td>
<td>1.000</td>
<td>2214.4</td>
</tr>
</tbody>
</table>

### Propane Section Equipment Vectors

<table>
<thead>
<tr>
<th></th>
<th>386.3</th>
<th>241.2</th>
<th>1.00</th>
<th>3.0</th>
<th>4.0</th>
<th>645852</th>
<th>1.000</th>
<th>11697</th>
<th>0.000</th>
<th>3509</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>10</td>
<td>3.0</td>
<td>0.0</td>
<td>5.0</td>
<td>0.0</td>
<td>39.0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>3.0</td>
<td>0.0</td>
<td>6.0</td>
<td>0.0</td>
<td>25.0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>5.0</td>
<td>0.0</td>
<td>7.0</td>
<td>0.0</td>
<td>555.0</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>P</td>
<td>Q</td>
<td>F</td>
<td>C</td>
<td>T</td>
<td>P</td>
<td>Q</td>
<td>F</td>
<td>C</td>
<td>T</td>
</tr>
<tr>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
<td>----</td>
</tr>
<tr>
<td>18.3</td>
<td>45.1</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>193344</td>
<td>23186</td>
<td>81189</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>21</td>
<td>7</td>
<td>6</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>45.1</td>
<td>45.1</td>
<td>0.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>11</td>
<td>8</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>1729</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**C. REFRIGERATION LEVEL DETAILS** - TEMP, PRES, DEMAND, FLOW, UNIT COST -

<table>
<thead>
<tr>
<th>T</th>
<th>P</th>
<th>Q</th>
<th>F</th>
<th>C</th>
<th>T</th>
<th>P</th>
<th>Q</th>
<th>F</th>
<th>C</th>
<th>T</th>
<th>P</th>
<th>Q</th>
<th>F</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>296.0</td>
<td>11.5</td>
<td>2400000</td>
<td>419.2</td>
<td>0.0000138</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>345.0</td>
<td>44.5</td>
<td>0</td>
<td>0</td>
<td>0.0000000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>385.0</td>
<td>106.9</td>
<td>1520000</td>
<td>273.4</td>
<td>0.0000095</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>425.0</td>
<td>18.3</td>
<td>5862117</td>
<td>1189.2</td>
<td>0.0000043</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>470.0</td>
<td>45.1</td>
<td>5670000</td>
<td>1025.2</td>
<td>0.0000025</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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  
$\text{SPARLIST}$  
$\text{AMORT}=0.3, \text{HRS}=8000,$  
$\text{TWAT}=535, \text{DIW}=10, \text{CWAT}=3.6E-5, \text{TS}=825, \text{HVS}=1.00E3, \text{CS}=1.80E-2, \text{CKWH}=0.007,$  
$\text{APPP}=15, \text{APRR}=10, \text{ARRR}=5, \text{TRRR}=310.$  
$\$END$

**C**. COLUMN NO.  
1 DEETHANIZER  
2 DEMETHANIZER  
3 HP C2 SPLITTER  
4 LP C2 SPLITTER  
5 DEPROPNIZER  
6 C3 SPLITTER  

**SKPMLST**  
$\text{NIS}=13, \text{NKPM}=6,$  
$\text{KPM1}=1,100,1,-2,-3,0,$  
$\text{KPM2}=2,100,2,-4,-5,0,$  
$\text{KPM3}=3,100,5,-6,-7,0,$  
$\text{KPM4}=4,100,7,-8,-9,0,$  
$\text{KPM5}=5,100,3,-10,-11,0,$  
$\text{KPM6}=6,100,10,-12,-13,0.$  
$\$END$

**$\text{SMPLST}$**  
$\text{SMPA1}=1,1,1,215,0,420,2*0,$  
$\text{SMPA2}=2,0,1,0,270,2*0,$  
$\text{SMPA9}=9,-1,1,0,2,0,2*0,$  
$\text{SMPB1}=2*0,520,115,0,1,1500,255,495,315,210,135,45,45,0.$  
$\$END$

**$\text{EMILST}$**  
$\text{EMI1}=1,100,1,1,2,0.162,0.0025,0.015,0.565,4,5,250,24,3*0,$  
$\text{EMI2}=2,100,-1,1,1,0.98,0.005,0.01,0.58,2,3,75,24,3*0,$  
$\text{EMI3}=3,100,0,1,1,0.96,0.03,0.45,0.54,3,4,65,24,3*0,$  
$\text{EMI4}=4,100,-1,1,1,0.96,0.04,0.01,0.94,3,4,20,24,3*0,$  
$\text{EMI5}=5,100,0,1,2,0.25,0.035,0.06,0.835,6,7,20,24,3*0,$  
$\text{EMI6}=6,100,0,1,2,0.9,0.08,0.075,0.76,5,6,115,18,3*0.$  
$\$END$
<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>0</td>
<td>1.00</td>
<td>396.5</td>
<td>0.73</td>
<td>11</td>
<td>9</td>
<td>3.8</td>
<td>0.60</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>14683</td>
<td>86854</td>
<td>0</td>
<td>26056</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>0</td>
<td>-1.00</td>
<td>0.0</td>
<td>0.27</td>
<td>9</td>
<td>7</td>
<td>2.31</td>
<td>0.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.56</td>
<td>1495</td>
<td>49368</td>
<td>0</td>
<td>14809</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>0</td>
<td>0.00</td>
<td>0.0</td>
<td>1.93</td>
<td>21</td>
<td>7</td>
<td>2.73</td>
<td>0.13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.72</td>
<td>2830</td>
<td>63500</td>
<td>0</td>
<td>19050</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>0</td>
<td>-1.00</td>
<td>0.0</td>
<td>2.49</td>
<td>17</td>
<td>22</td>
<td>2.33</td>
<td>0.09</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.72</td>
<td>3440</td>
<td>104832</td>
<td>0</td>
<td>31450</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>0</td>
<td>0.00</td>
<td>0.0</td>
<td>0.48</td>
<td>7</td>
<td>10</td>
<td>1.91</td>
<td>0.27</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>2793</td>
<td>24773</td>
<td>0</td>
<td>7432</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>100</td>
<td>0</td>
<td>0.00</td>
<td>0.0</td>
<td>6.14</td>
<td>29</td>
<td>56</td>
<td>3.93</td>
<td>0.31</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.00</td>
<td>21176</td>
<td>228661</td>
<td>0</td>
<td>68598</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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=14., CWAT=3.6E-5, TS=825., HVS=1.00E3, CS=1.80E-2, CKWH=0.007,
APPR=10., APRR=10., ARRR=5., TRRR=310.,
$END

C.. HOT STREAM DATA

<table>
<thead>
<tr>
<th>1</th>
<th>-2</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>555.2</td>
<td>555.2</td>
<td>610.0</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>240.0</td>
<td>449.1</td>
<td>520.0</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>257.90</td>
<td>583.15</td>
<td>684.32</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>510.6</td>
<td>513.6</td>
<td>513.6</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

C.. COLD STREAM DATA

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>583.8</td>
<td>583.8</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>644.3</td>
<td>644.3</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>533.6</td>
<td>533.6</td>
</tr>
<tr>
<td>0.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

C.. REFRIGERATION LEVELS -

| 305. | 345. | 385. | 425. | 470. |
### C*** LPM OPTIMUM PLANT ***

#### C. HOT STREAM DATA

<table>
<thead>
<tr>
<th></th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>1</td>
<td>174.4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>555.0</td>
</tr>
<tr>
<td>2</td>
<td>555.2</td>
<td>555.2</td>
<td>610.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6580</td>
<td>1000</td>
<td>1</td>
<td>420.0</td>
</tr>
<tr>
<td>3</td>
<td>240.0</td>
<td>449.1</td>
<td>520.0</td>
<td>0</td>
<td>0</td>
<td>115.0</td>
<td>5185.628</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>240.0</td>
<td>423.9</td>
<td>423.0</td>
<td>0</td>
<td>0</td>
<td>250.0</td>
<td>4567941</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>553.9</td>
<td>560.4</td>
<td>560.4</td>
<td>0</td>
<td>0</td>
<td>200.0</td>
<td>1229635</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>510.6</td>
<td>513.6</td>
<td>513.6</td>
<td>0</td>
<td>0</td>
<td>115.0</td>
<td>3810960</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1000</td>
<td>0</td>
<td>0</td>
<td>250.0</td>
<td>0</td>
<td>420.0</td>
</tr>
<tr>
<td>8</td>
<td>250.0</td>
<td>484.9</td>
<td>620.6</td>
<td>0</td>
<td>0</td>
<td>250.0</td>
<td>6743676</td>
<td>1000</td>
<td>1000</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>240.0</td>
<td>484.9</td>
<td>545.0</td>
<td>0</td>
<td>0</td>
<td>250.0</td>
<td>5417232</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>240.0</td>
<td>484.9</td>
<td>495.0</td>
<td>0</td>
<td>0</td>
<td>250.0</td>
<td>4578582</td>
<td>1000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>11</td>
<td>250.0</td>
<td>484.9</td>
<td>445.0</td>
<td>0</td>
<td>0</td>
<td>250.0</td>
<td>2407796</td>
<td>851.0</td>
<td>1500</td>
<td>0</td>
</tr>
</tbody>
</table>

#### C. COLD STREAM DATA

<table>
<thead>
<tr>
<th></th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>250.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>583.8</td>
<td>603.6</td>
<td>583.8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-346934</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>644.3</td>
<td>653.7</td>
<td>644.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>702280</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>533.6</td>
<td>547.4</td>
<td>533.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1297719</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

#### C. EQUIPMENT DATA

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th></th>
<th>2</th>
<th>0</th>
<th>0</th>
<th>6</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>680.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>228460</td>
<td>28422</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>6</td>
<td>202</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1326444</td>
<td>1</td>
<td>18337</td>
</tr>
<tr>
<td>3</td>
<td>155</td>
<td>13.4</td>
<td>7369.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6762353</td>
<td>1000</td>
<td>7285</td>
<td>54099</td>
</tr>
<tr>
<td>4</td>
<td>56.9</td>
<td>47.2</td>
<td>92.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1669474</td>
<td>1</td>
<td>3707</td>
<td>13356</td>
</tr>
<tr>
<td>5</td>
<td>7</td>
<td>16.0</td>
<td>396.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4937518</td>
<td>2.0</td>
<td>3914.8</td>
<td>387120</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>7</td>
<td>203</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1594803</td>
<td>1000</td>
<td>19881</td>
<td>2552</td>
</tr>
<tr>
<td>7</td>
<td>11</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>127886</td>
<td>13971</td>
<td>52337</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>8</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>6608847</td>
<td>1000</td>
<td>68191</td>
</tr>
<tr>
<td>9</td>
<td>13</td>
<td>149.0</td>
<td>7</td>
<td>203</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>-838650</td>
<td>1000</td>
<td>22140</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>490.0</td>
<td>21.0</td>
<td>445.0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>-2170786</td>
<td>1000</td>
<td>14187</td>
</tr>
<tr>
<td>11</td>
<td>16</td>
<td>238.0</td>
<td>28.0</td>
<td>203.0</td>
<td>10</td>
<td>0</td>
<td>0</td>
<td>-1382487</td>
<td>2000</td>
<td>8757</td>
</tr>
</tbody>
</table>
### LP PLANT - LOW TEMP SECTION - NEW 2/1 MATCH SET

**$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
- ARR = 7
- TRRR = 400

**$END**

#### HOT STREAM DATA

<table>
<thead>
<tr>
<th>Stream</th>
<th>Data 1</th>
<th>Data 2</th>
<th>Data 3</th>
<th>Data 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>555.2</td>
<td>555.2</td>
<td>610.0</td>
<td>555.0</td>
</tr>
<tr>
<td>2</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>240.0</td>
<td>396.5</td>
<td>396.5</td>
<td>250.0</td>
</tr>
<tr>
<td>4</td>
<td>255.0</td>
<td>495.0</td>
<td>314.62</td>
<td>350.1</td>
</tr>
</tbody>
</table>

#### COLD STREAM DATA

<table>
<thead>
<tr>
<th>Stream</th>
<th>Data 1</th>
<th>Data 2</th>
<th>Data 3</th>
<th>Data 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>373.3</td>
<td>384.3</td>
<td>373.3</td>
<td>75.0</td>
</tr>
<tr>
<td>2</td>
<td>379.1</td>
<td>386.0</td>
<td>379.1</td>
<td>0.00</td>
</tr>
<tr>
<td>3</td>
<td>342.7</td>
<td>345.8</td>
<td>342.7</td>
<td>307.12</td>
</tr>
<tr>
<td>4</td>
<td>342.7</td>
<td>345.8</td>
<td>342.7</td>
<td>428.20</td>
</tr>
<tr>
<td>5</td>
<td>240.0</td>
<td>240.0</td>
<td>250.0</td>
<td>194.55</td>
</tr>
</tbody>
</table>

#### REFRIGERATION LEVELS

- 245
- 275
- 305
- 345
- 385
- 425
- 470
### HOT STREAM DATA

<table>
<thead>
<tr>
<th></th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>1.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>555.2</td>
<td>555.2</td>
<td>627.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>240.0</td>
<td>396.5</td>
<td>396.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>350.1</td>
<td>362.1</td>
<td>362.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>315.1</td>
<td>316.1</td>
<td>316.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>351.1</td>
<td>356.1</td>
<td>356.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>240.0</td>
<td>396.5</td>
<td>380.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>385.9</td>
<td>395.9</td>
<td>426.6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>385.9</td>
<td>395.9</td>
<td>388.9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>350.1</td>
<td>362.1</td>
<td>345.9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>351.1</td>
<td>359.5</td>
<td>402.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>240.0</td>
<td>396.5</td>
<td>350.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>240.0</td>
<td>396.5</td>
<td>351.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### COLD STREAM DATA

<table>
<thead>
<tr>
<th></th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>3401</th>
<th>6529</th>
<th>0.0069</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
<th>0.0000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>75.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>373.3</td>
<td>384.3</td>
<td>373.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>279.1</td>
<td>386.0</td>
<td>379.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>342.7</td>
<td>345.8</td>
<td>342.7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>342.7</td>
<td>345.8</td>
<td>343.1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>342.7</td>
<td>345.8</td>
<td>345.8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### EQUIPMENT DATA

<table>
<thead>
<tr>
<th></th>
<th>1.30</th>
<th>-4</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>343.0</td>
<td>535.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>2</td>
<td>2.30</td>
<td>-5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>250.0</td>
<td>535.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>5</td>
<td>11.0</td>
<td>2</td>
<td>-1</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>7</td>
<td>65.0</td>
<td>132.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>8</td>
<td>2725.8</td>
<td>7.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>14</td>
<td>20.0</td>
<td>55.2</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>15</td>
<td>1.10</td>
<td>4</td>
<td>-3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

---

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

---

253
<table>
<thead>
<tr>
<th>915.2</th>
<th>10.3</th>
<th>0.0</th>
<th>0.0</th>
<th>2.0</th>
<th>46800</th>
<th>0</th>
<th>14040</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>58.6</td>
<td>301</td>
<td>-6</td>
<td>0</td>
<td>0</td>
<td>-1091359</td>
<td>0</td>
</tr>
<tr>
<td>22</td>
<td>811.1</td>
<td>16.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2.0</td>
<td>7546</td>
</tr>
<tr>
<td>23</td>
<td>45.0</td>
<td>16.0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1689628</td>
<td>0</td>
</tr>
<tr>
<td>30</td>
<td>479.0</td>
<td>15.0</td>
<td>203</td>
<td>0</td>
<td>0</td>
<td>-234684</td>
<td>2.0</td>
</tr>
<tr>
<td>34</td>
<td>276.0</td>
<td>19.0</td>
<td>203</td>
<td>0</td>
<td>0</td>
<td>-1395937</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>66.0</td>
<td>270.0</td>
<td>0</td>
<td>0</td>
<td>-1507047</td>
<td>3.5</td>
</tr>
</tbody>
</table>
### C**-ll-** LP REFRIGERATION UNIT ***

#### C. METHANE SECTION STREAM VECTORS

<table>
<thead>
<tr>
<th></th>
<th>240.0</th>
<th>250.0</th>
<th>315.0</th>
<th>315.0</th>
<th>400.0</th>
<th>-1070</th>
<th>0.000</th>
<th>1.000</th>
<th>749.7</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>240.0</td>
<td>240.0</td>
<td>315.0</td>
<td>315.0</td>
<td>400.0</td>
<td>-1008000</td>
<td>0.000</td>
<td>678.0</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>240.0</td>
<td>240.0</td>
<td>315.0</td>
<td>315.0</td>
<td>400.0</td>
<td>-1195000</td>
<td>0.000</td>
<td>678.0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>240.0</td>
<td>240.0</td>
<td>315.0</td>
<td>295.0</td>
<td>75.0</td>
<td>1236000</td>
<td>1.000</td>
<td>749.7</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>240.0</td>
<td>240.0</td>
<td>315.0</td>
<td>295.0</td>
<td>75.0</td>
<td>-1195000</td>
<td>0.000</td>
<td>678.0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>240.0</td>
<td>240.0</td>
<td>295.0</td>
<td>295.0</td>
<td>75.0</td>
<td>547000</td>
<td>1.000</td>
<td>572.0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>240.0</td>
<td>240.0</td>
<td>295.0</td>
<td>295.0</td>
<td>75.0</td>
<td>1014000</td>
<td>1.000</td>
<td>106.0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>240.0</td>
<td>240.0</td>
<td>295.0</td>
<td>295.0</td>
<td>75.0</td>
<td>547000</td>
<td>1.000</td>
<td>572.0</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>240.0</td>
<td>240.0</td>
<td>295.0</td>
<td>295.0</td>
<td>75.0</td>
<td>1014000</td>
<td>1.000</td>
<td>106.0</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>240.0</td>
<td>240.0</td>
<td>295.0</td>
<td>295.0</td>
<td>75.0</td>
<td>547000</td>
<td>1.000</td>
<td>572.0</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>240.0</td>
<td>240.0</td>
<td>295.0</td>
<td>295.0</td>
<td>75.0</td>
<td>1014000</td>
<td>1.000</td>
<td>106.0</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>240.0</td>
<td>240.0</td>
<td>295.0</td>
<td>295.0</td>
<td>75.0</td>
<td>547000</td>
<td>1.000</td>
<td>572.0</td>
<td></td>
</tr>
</tbody>
</table>

#### C. METHANE SECTION EQUIPMENT VECTORS

<table>
<thead>
<tr>
<th></th>
<th>210.0</th>
<th>27.0</th>
<th>0.00</th>
<th>6.00</th>
<th>85.0</th>
<th>110.0</th>
<th>0.00</th>
<th>106.0</th>
<th>106.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>210.0</td>
<td>27.0</td>
<td>0.00</td>
<td>6.00</td>
<td>85.0</td>
<td>110.0</td>
<td>0.00</td>
<td>106.0</td>
<td>106.0</td>
</tr>
<tr>
<td>3</td>
<td>435.0</td>
<td>435.0</td>
<td>352.7</td>
<td>352.7</td>
<td>263.9</td>
<td>-7828432</td>
<td>0.000</td>
<td>594.1</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>435.0</td>
<td>435.0</td>
<td>352.7</td>
<td>352.7</td>
<td>263.9</td>
<td>-7828432</td>
<td>0.000</td>
<td>594.1</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>435.0</td>
<td>435.0</td>
<td>352.7</td>
<td>352.7</td>
<td>263.9</td>
<td>-7828432</td>
<td>0.000</td>
<td>594.1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>435.0</td>
<td>435.0</td>
<td>352.7</td>
<td>352.7</td>
<td>263.9</td>
<td>-7828432</td>
<td>0.000</td>
<td>594.1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>435.0</td>
<td>435.0</td>
<td>352.7</td>
<td>352.7</td>
<td>263.9</td>
<td>-7828432</td>
<td>0.000</td>
<td>594.1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>435.0</td>
<td>435.0</td>
<td>352.7</td>
<td>352.7</td>
<td>263.9</td>
<td>-7828432</td>
<td>0.000</td>
<td>594.1</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>435.0</td>
<td>435.0</td>
<td>352.7</td>
<td>352.7</td>
<td>263.9</td>
<td>-7828432</td>
<td>0.000</td>
<td>594.1</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>435.0</td>
<td>435.0</td>
<td>352.7</td>
<td>352.7</td>
<td>263.9</td>
<td>-7828432</td>
<td>0.000</td>
<td>594.1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>435.0</td>
<td>435.0</td>
<td>352.7</td>
<td>352.7</td>
<td>263.9</td>
<td>-7828432</td>
<td>0.000</td>
<td>594.1</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>435.0</td>
<td>435.0</td>
<td>352.7</td>
<td>352.7</td>
<td>263.9</td>
<td>-7828432</td>
<td>0.000</td>
<td>594.1</td>
<td></td>
</tr>
</tbody>
</table>

#### C. ETHYLENE SECTION STREAM VECTORS

<table>
<thead>
<tr>
<th></th>
<th>240.0</th>
<th>240.0</th>
<th>305.0</th>
<th>305.0</th>
<th>200.0</th>
<th>-933912</th>
<th>0.000</th>
<th>199.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>342.7</td>
<td>342.7</td>
<td>342.7</td>
<td>342.7</td>
<td>263.9</td>
<td>-6552384</td>
<td>0.000</td>
<td>173.5</td>
</tr>
<tr>
<td>3</td>
<td>435.0</td>
<td>435.0</td>
<td>390.3</td>
<td>390.3</td>
<td>263.9</td>
<td>-6552384</td>
<td>0.000</td>
<td>173.5</td>
</tr>
<tr>
<td>4</td>
<td>435.0</td>
<td>435.0</td>
<td>390.3</td>
<td>390.3</td>
<td>263.9</td>
<td>-6552384</td>
<td>0.000</td>
<td>173.5</td>
</tr>
<tr>
<td>5</td>
<td>435.0</td>
<td>435.0</td>
<td>390.3</td>
<td>390.3</td>
<td>263.9</td>
<td>-6552384</td>
<td>0.000</td>
<td>173.5</td>
</tr>
<tr>
<td>6</td>
<td>435.0</td>
<td>435.0</td>
<td>390.3</td>
<td>390.3</td>
<td>263.9</td>
<td>-6552384</td>
<td>0.000</td>
<td>173.5</td>
</tr>
<tr>
<td>7</td>
<td>435.0</td>
<td>435.0</td>
<td>390.3</td>
<td>390.3</td>
<td>263.9</td>
<td>-6552384</td>
<td>0.000</td>
<td>173.5</td>
</tr>
<tr>
<td>8</td>
<td>435.0</td>
<td>435.0</td>
<td>390.3</td>
<td>390.3</td>
<td>263.9</td>
<td>-6552384</td>
<td>0.000</td>
<td>173.5</td>
</tr>
<tr>
<td>9</td>
<td>435.0</td>
<td>435.0</td>
<td>390.3</td>
<td>390.3</td>
<td>263.9</td>
<td>-6552384</td>
<td>0.000</td>
<td>173.5</td>
</tr>
<tr>
<td>10</td>
<td>435.0</td>
<td>435.0</td>
<td>390.3</td>
<td>390.3</td>
<td>263.9</td>
<td>-6552384</td>
<td>0.000</td>
<td>173.5</td>
</tr>
<tr>
<td>11</td>
<td>435.0</td>
<td>435.0</td>
<td>390.3</td>
<td>390.3</td>
<td>263.9</td>
<td>-6552384</td>
<td>0.000</td>
<td>173.5</td>
</tr>
<tr>
<td>12</td>
<td>435.0</td>
<td>435.0</td>
<td>390.3</td>
<td>390.3</td>
<td>263.9</td>
<td>-6552384</td>
<td>0.000</td>
<td>173.5</td>
</tr>
</tbody>
</table>

#### C. ETHYLENE SECTION EQUIPMENT VECTORS

<table>
<thead>
<tr>
<th></th>
<th>1162.8</th>
<th>34.0</th>
<th>0.00</th>
<th>6.00</th>
<th>3.50</th>
<th>99194</th>
<th>0.000</th>
<th>29758</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1162.8</td>
<td>34.0</td>
<td>0.00</td>
<td>6.00</td>
<td>3.50</td>
<td>99194</td>
<td>0.000</td>
<td>29758</td>
</tr>
<tr>
<td>3</td>
<td>263.9</td>
<td>15.2</td>
<td>0.00</td>
<td>8.00</td>
<td>0.00</td>
<td>11.00</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>263.9</td>
<td>15.2</td>
<td>0.00</td>
<td>8.00</td>
<td>0.00</td>
<td>11.00</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>263.9</td>
<td>15.2</td>
<td>0.00</td>
<td>8.00</td>
<td>0.00</td>
<td>11.00</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>263.9</td>
<td>15.2</td>
<td>0.00</td>
<td>8.00</td>
<td>0.00</td>
<td>11.00</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>
256.9 106.9 0.0 0.0 0.0 0.0 0.0
7 11 10 0.0 13 0 0 201
15.2 44.5 0.0
8 21 13 11 0.0 14 0 0 100
44.5 44.5 0.0
9 11 14 0.0 15 0 0 252
44.5 106.9 0.0
10 21 15 0.0 16 0 0 100
106.9 106.9 12 100
11 263.9 16 0.0 17 0 0 600
106.9 263.9 0.0 0.0 0.0 0.0 0.0
12 203.9 425.1 100 118084 0.0 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
3 555.0 555.0 555.0 174.4 174.4 12410690 0.000 2624.1
4 555.0 555.0 546.5 174.4 174.4 12410690 0.000 2624.1
5 240.0 240.0 545.0 75.0 2506716 1.000 749.7
6 555.0 555.0 540.7 174.4 174.4 12410690 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 908336 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 49.1 11292594 1.000 2626.9
12 555.2 555.2 626.9 174.4 174.4 16976226 1.000 2476.1

C. PROPANE FLOW TO PSEUDO-SERVICE - 150.8

C. PROPANE SECTION EQUIPMENT VECTORS -
1 386.4 25.7 1 0.0 4 5 -690193 1.000 11698 0.0 3510
1 197.7 51.4 2 0.0 6 7 -462944 2.00 15651 0.0 4695
1 174.4 18.3 0.0 0.0 8 0 0.0 41 0.0 0.0 0.0
1 174.4 45.1 0.0 0.0 9 0 0.0 27 0.0 0.0 0.0
1 18.3 45.1 0.0 0.0 10 0 0.0 0.0 0.0 0.0 0.0
1 1153 352786 48167 154003
1 6.2 10 0.0 9 0.0 10 0 0.0 0.0 0.0 0.0 0.0
1 100 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

C. PROPANE SECTION EQUIPMENT VECTORS -
1 9162.9 7.2 202 103675.4 -18661564 1.000 147784 29859 74194

256
REFERENCES FOR THE APPENDICES

1. Lee, K. F., Masso, A. H. and Rudd, D. F.,


3. Hengstebeck, R. J.,

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

5. Bolles, W. L.,

6. Peters, M. S. and Timmerhaus, K. D.,

7. Edmister, W. C.,