

ORNL/Sub/86-XSY123V

**MODULAR SIMULATION  
OF ABSORPTION SYSTEMS**

**USER'S GUIDE AND REFERENCE**

WINDOWS Version 5.0 (**AbsimW**) March 1998

April 2000

This code was developed under the sponsorship of an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UT-Battelle, LLC, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or UT-Battelle, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof nor those of UT-Battelle, LLC, and shall not be used for advertising or product endorsement purposes.

**A B S I M**

**MODULAR SIMULATION OF ABSORPTION SYSTEMS**

**USER'S GUIDE AND REFERENCE**

WINDOWS Version 5.0 (**AbsimW**) March 1998

Prepared by Gershon Grossman  
for the  
OAK RIDGE NATIONAL LABORATORY  
Oak Ridge, Tennessee 38931  
managed by  
UT-Battelle, LLC  
for the  
U.S. DEPARTMENT OF ENERGY  
under Contract No. DE-AC05-00OR22725

## **ACKNOWLEDGEMENTS**

This code was developed by Dr. Gershon Grossman for the Oak Ridge National Laboratory. Funding for this program was obtained under the U.S. Department of Energy Building Equipment Technology Program.

# CONTENTS

Page

1.	OVERVIEW.....	1
1.1	Welcome to ABSIM.....	1
1.2	Background .....	2
2.	GETTING STARTED.....	5
2.1	Installing ABSIMW on your Computer .....	5
2.2	Running ABSIMW .....	6
3.	STRUCTURE OF THE COMPUTER MODEL.....	9
3.1	The Input .....	10
3.2	The Main Program .....	12
3.3	The Output.....	12
4.	UNIT SUBROUTINES AND GOVERNING EQUATIONS .....	15
4.1	Absorber (Subroutine ABSORB).....	19
4.2	Desorber (Subroutine DESORB) .....	20
4.3	Heat Exchanger (Subroutine HEX).....	22
4.4	Condenser (Subroutine COND) .....	23
4.5	Evaporator (Subroutine EVAP) .....	24
4.6	Valve (Subroutine VALVE).....	25
4.7	Mixer (Subroutine MIX) .....	26
4.8	Splitter (Subroutine SPLIT) .....	27
4.9	Rectifier (Subroutine RECT) .....	27
4.10	Analyser (Subroutine ANALYS) .....	29
4.11	Compressor (Subroutine COMP).....	31
4.12	Pump (subroutine PUMP).....	31
5.	MATERIAL PROPERTIES .....	33
6.	METHOD OF SOLUTION .....	35
7.	USING THE GRAPHICAL INTERFACE .....	39
7.1	File.....	39
7.2	Add.....	41

7.3	Data .....	41
7.4	Results .....	43
7.5	Plot .....	44
7.6	Renumber .....	45
7.7	Utilities .....	45
7.8	Help .....	46
7.9	Tool Box.....	46
8.	RESULTS OF SIMULATION.....	49
9.	REFERENCES.....	51
	NOMENCLATURE.....	53
	Appendix A: INPUT MANUAL.....	55
	A.1. General Information .....	55
	A.2. Unit Input .....	57
	A.3. State Point Input.....	59
	Appendix B: SAMPLE CYCLES.....	61

# 1. OVERVIEW

## 1.1 WELCOME TO ABSIM

ABSIM (an acronym for ABSorption SIMulation) is a user-oriented computer code designed for the simulation of absorption systems at steady state, in both flexible and modular form. ABSIM makes it possible to investigate various cycle configurations with different working fluids, to calculate their operating parameters, to predict their performance and to compare them with each other on a uniform basis. A graphical user interface enables the user to draw the cycle diagram on the computer screen, enter data interactively, run the program and view the results either in the form of a table or superimposed on the cycle diagram. Special utilities enable the user to plot the results and produce a pressure-temperature-concentration ( P-T-X) diagram of the cycle.

Most absorption systems consist of a number of standard components or units (e.g., absorber, condenser) that may be combined in different forms to produce various cycles. Recognizing this, ABSIM has been structured around unit subroutines, each of which contains the governing equations for the particular unit. These subroutines are activated by a main program that interprets the input for the cycle, calls the units, and links them to each other in an order corresponding to the user's specification to form the complete system. Each unit subroutine, when activated, addresses a property database for the thermodynamic properties of the working fluids. The equations generated by the code are listed and solved simultaneously by a mathematical solver routine.

The code requires relatively simple inputs, consisting of the minimum information needed to define an absorption system properly. After drawing the cycle in terms of the units recognizable by the code and showing their interconnections, the user must specify the size of each exchange unit in terms of its heat and mass transfer characteristics, the working fluid(s) at each state point; and the given operating conditions, such as temperatures, flowrates, and the like, fixed at specific state points. Based on this information, the program calculates the temperature, flowrate, concentration, pressure, and the vapor fraction at each state point in the system and the heat duty at each unit, from which the coefficient of performance may be determined.

ABSIM has been used successfully to simulate a variety of single-, double- and triple-stage absorption chillers, heat pumps and heat transformers using the working fluids LiBr-H<sub>2</sub>O, H<sub>2</sub>O-NH<sub>3</sub>, LiBr/H<sub>2</sub>O-NH<sub>3</sub>, LiBr/ZnBr<sub>2</sub>-CH<sub>3</sub>OH and more. Some of these results will be described briefly in Sect. 8. Eleven absorption fluids are presently available in the code's property database, and 12 units are available to compose practically every absorption cycle of interest. The code in its present form may be used not only to evaluate new cycles and working fluids, but also to

investigate a system's behavior in off-design conditions, to analyze experimental data, and to perform preliminary design optimization.

This user manual is organized into eight sections and two appendices. The remainder of this section describes the background for the ABSIM code and presents its special features in comparison with other simulation codes. Section 2 contains information on installation of the code and on basic operations for the first-time user. Section 3 describes the structure of the code, including the input, the output, and the main program. Section 4 describes the unit subroutines containing the governing equations for the 12 unit modules of the code. Section 5 reviews the property database that contains the thermodynamic properties of the working fluids. Section 6 describes the solver package and the method of solution for the equations generated by the code. Section 7 instructs the user on how to use the graphical interface. Some results of the simulation are described in Sect. 8. Appendix A is an input manual describing in detail each item in the input, its significance, and its format. Appendix B contains cycle diagrams and input and output files for several sample cycles provided with the code.

## **1.2 BACKGROUND**

Absorption heat pumps, once dominant in the cooling industry, have received renewed and growing attention in the past two decades. The increasing cost of electricity has made the particular features of this heat-powered cycle attractive for both residential and industrial applications. Solar-powered absorption chillers, gas-fired domestic heat pumps, and waste-heat-powered industrial temperature boosters are a few of the applications that have become the subject of intensive research and development. Some of these developments have matured into commercial products and are competing successfully in the market with their electric-powered counterparts. The "absorption community" has begun to search for advanced cycles in various multistage configurations, and for new working fluid combinations with potential for enhanced performance and reliability.

The development of working absorption systems has created a need for reliable and effective system simulations. Several system-specific computer models have been developed [1–5] that have proved to be very valuable tools for both research and development and design optimization. Some of them were validated against experimental data with good agreement. System-specific models are, however, quite restricted. All the above models were limited to the particular system for which they were created; their structure did not allow easy modification to model other systems. For that to be done, major parts of the code, particularly the iterative sequence, would have to be rewritten by the user for each new cycle.



In the mid-1980s and early 1990s, under the U.S. Department of Energy Absorption Program, a wide variety of advanced cycles and working fluids were screened [6–8] in order to select from among them promising candidates for further development. As part of this effort, attempts were made to compare different candidate cycles using software used by the chemical industry for process simulation [9]. Several commercially available codes are designed to simulate large chemical plants and processes in a flexible form. These codes contain extensive libraries of modules and working substances. Experience has shown that, for simulating common heat pump cycles, these codes are excessively powerful. At the same time, and perhaps because of their all-encompassing approach, they are not user-friendly and require of the user considerable investment in time and training before they can be used. Further, the extensive libraries containing many modules and working substances do not contain most of those required for absorption systems; these modules must be introduced by the user.

There are two main approaches to process simulation: one is referred to in the literature as “sequential,” the other as “equations-oriented.” Codes using the sequential algorithms (e.g., [10]) require the user to program the sequence by which the simulator is to proceed through the cycle. This programming is easy to do in the open cycles most often used in the chemical industry, but when several interconnected loops are encountered in a closed cycle, one can run into convergence problems. In the equations-oriented codes (e.g., [11]) the user must write the equations governing the cycle (or obtain them from library modules) and make sure the set is well-defined. Furthermore, the user must provide a good initial guess for all the unknowns, and in the case of a nonlinear set of equations, check the results to make sure that the mathematical solution is also a physical one. In one form or another, the user has to do a fair amount of programming to lead the simulator toward convergence to the correct solution.

The objective of ABSIM has been to provide a modular, user-oriented computer simulation code, specifically designed for simulating absorption systems in different cycle configurations and with different working fluids. The usefulness of such a code has emerged during repeated attempts under absorption research programs to evaluate new ideas for absorption cycles and working substances, and to compare them with existing ones. In developing the code, care has been taken to avoid most of the above-mentioned requirements imposed on the user in common process simulators. One particular feature of ABSIM is that *the user does not have to do any programming*, which enhances user-friendliness considerably. The complete set of governing equations is generated automatically by simply drawing the cycle diagram on the computer screen, thereby specifying to the code the components of the system and their interconnections. The set is solved in a manner that eliminates mathematically valid results that are not also physically valid. The following sections will describe the modular approach taken to achieve this

flexibility without the need for the user to worry about the iterative scheme, or even see the equations.

## 2. GETTING STARTED

The Windows version of ABSIM (AbsimW) is distributed on a standard 3.5-in. diskette containing the program files in the root directory, and some sample input files in the subdirectory \DATA. The root directory contains the following files:

ABSIMW.EXE  
ABSIMW.CFG  
ABSIMW.HLP  
UNITS.GEO  
SIMPLE.VF  
FLUID.TYP  
START.TXT  
ABSIM.EXE  
ABSIM1.EXE  
ABSIM1.PIF  
README.DOC  
\DATA  
\DLL

AbsimW is designed to run in the Windows 95 or Windows 3.11 environment, on IBM-compatible personal computers (486 or higher) equipped with a hard disk and a mouse.

### 2.1 INSTALLING ABSIMW ON YOUR COMPUTER

To run AbsimW, you must copy the diskette to the hard disk on your computer. First, create on the hard disk a new directory by a name of your choice (e.g., ABSIMW) and copy the contents of the diskette to it. Then create a subdirectory by the name DATA under the first directory (e.g., ABSIMW\DATA) and copy the files from the diskette subdirectory \DATA into it. To configure your system for your choice of name, edit the file ABSIMW.CFG: Change the one line of the file to be the full path of your \DATA directory.

The two DLL files contained in the diskette subdirectory \DLL are used by AbsimW to generate certain graphical features. During installation, copy these two files, BWCC.DLL and CTL3DV2.DLL, into the subdirectory \WINDOWS\SYSTEM on your hard disk.

Example: To install the program from disk drive A:\ onto hard drive C:\ under directory name ABSIMW, do the following:

```
C:\>  
C:\>mkdir ABSIMW  
C:\>cd ABSIMW  
C:\ABSIMW>mkdir DATA  
C:\ABSIMW>A:  
A:\>copy *.* C:\ABSIMW  
A:\>cd DATA  
A:\DATA>copy *.* C:\ABSIMW\DATA  
A:\DATA >C:  
C:\ABSIMW>
```

Now edit the file C:\ABSIMW\ABSIMW.CFG and change the line to

```
c:\absimw\data
```

You are now ready to run the program! From Windows, execute the file ABSIMW.EXE by clicking on the associated icon. A title window will appear briefly, followed by a disclaimer. Click on the OK box to acknowledge and obtain the main menu.

## 2.2 RUNNING ABSIMW

The main menu includes the following entries:

FILE – to manipulate ABSIM files

ADD – to add units to the cycle

DATA – to specify the operating conditions of the cycle

RESULTS – to view the results on the cycle diagram

PLOT – to plot certain calculated parameters vs others

RENUMBER – to change the number assigned to a unit or to a state point

UTILITIES – to shape the cycle diagram, change text size and form

HELP – to receive general instructions on ABSIM

AbsimW further contains a TOOL BOX (shown on the right hand side) to assist in drawing or modifying the drawing of a cycle.

More specific instructions on each item of the main menu are given in Sect. 7 describing the graphical interface for AbsimW. However, the program is quite user-friendly, and most of the entries are self-explanatory. At this point, you are encouraged to explore on your own and

attempt to perform some of the above operations. Begin by opening an existing cycle from the samples provided with the program by selecting FILE from the main menu and OPEN under FILE. Run the cycle by selecting RUN under FILE and view the results by selecting RESULTS. Then you may proceed to create your own simple cycle by selecting FILE from the main menu and NEW under FILE, then selecting ADD from the main menu. ADD various units and use the TOOL BOX to move, rotate, flip and align them, connect them to each other and RENUMBER them. SAVE your file, select DATA from the main menu and proceed to specify operating conditions, working fluids, etc. for your cycle, then attempt to RUN it. HELP from the main menu will guide you, but sooner or later you are bound to run into difficulties, which may induce you to read the remainder of this manual!

The following four sections describe the structure of the computer model, the unit subroutines, the property database and the solver. **If you want to continue to work with the program without attempting to fully fathom these details, you may skip to Sect. 7, which describes the graphical interface.**

### 3. STRUCTURE OF THE COMPUTER MODEL

The objective of developing a flexible and user-oriented simulation code has led to several basic requirements with respect to its structure. First, the code had to be modular in nature to enable the user to specify different cycle configurations, different working fluids, and obviously different unit sizes and operating conditions. Second, in order to keep the input requirements simple and the application straightforward, the code had to contain, as an integral part, all the equations and algorithms required for the solution. It was decided early in the development of ABSIM to eliminate the need for the user to write parts of the code specific to his or her cycle, in the form of external subroutines or otherwise, as is often done in sequential flowsheeting simulators. This is so because in a closed cycle containing several loops, defining a stable sequence of iteration can constitute a major difficulty for the user.

The structure of the code is based on the following logic: Recognizing that each absorption system consists of a number of standard components (e.g., absorber, evaporator, desorber, condenser), and that each of those components can use a variety of working fluids, the code is built upon unit subroutines and property subroutines in separate modules. Each basic component is simulated by a unit subroutine providing a mathematical expression of the physics of that component. The unit subroutine contains all the physical equations required to fully describe its behavior, such as energy balance, conservation of mass for each material species, heat and mass transfer, and thermodynamic equilibrium. When activated, the unit subroutine calls upon the property subroutines, contained in a property database, for the thermodynamic properties of the working fluids. This separation between the unit subroutines and property subroutines provides the flexibility for each unit to operate with different working fluids. The main program calls the unit subroutines and links the components together in a form corresponding to the user's specification of the cycle. Each call to a unit subroutine is equivalent to collecting all the equations associated with it, without attempting to solve them as yet. When the calls to all the unit subroutines have been completed, all the equations have been established and a mathematical solver routine is used to solve the set of nonlinear equations simultaneously. During the solution process, the solver calls the unit subroutines to calculate the residual values of the equations for the current values of the variables.

Figure 1 is a schematic description of the program structure. User input conveys to the program an 'image' of the cycle to be simulated: the number and types of units contained in it, their interconnections, and size or transfer characteristics, where applicable. The input must also specify the working fluids and contain the values of the parameters set fixed by the user (e.g., temperatures, flowrates) at specific state points. The main program interprets the input and creates a variable vector containing all the independent unknown quantities. It then calls the unit

subroutines, thereby establishing the system's governing equations. The unit subroutines refer to the property database as illustrated. All the equations and variables are normalized and brought to the same order of magnitude. The solver package is then activated, and the values of the unknowns are calculated to a user-specified accuracy. Recognizing that the set of nonlinear equations may have more than one mathematical solution, the validity of the solution is ensured by applying linear constraints on the variables that define a physically meaningful region. The output contains the temperature, enthalpy, flowrate, concentration, pressure, and vapor fraction at each of state point in the cycle, as well as the heat duty and transfer characteristics of each unit.

ORNL 2000-00032ldg

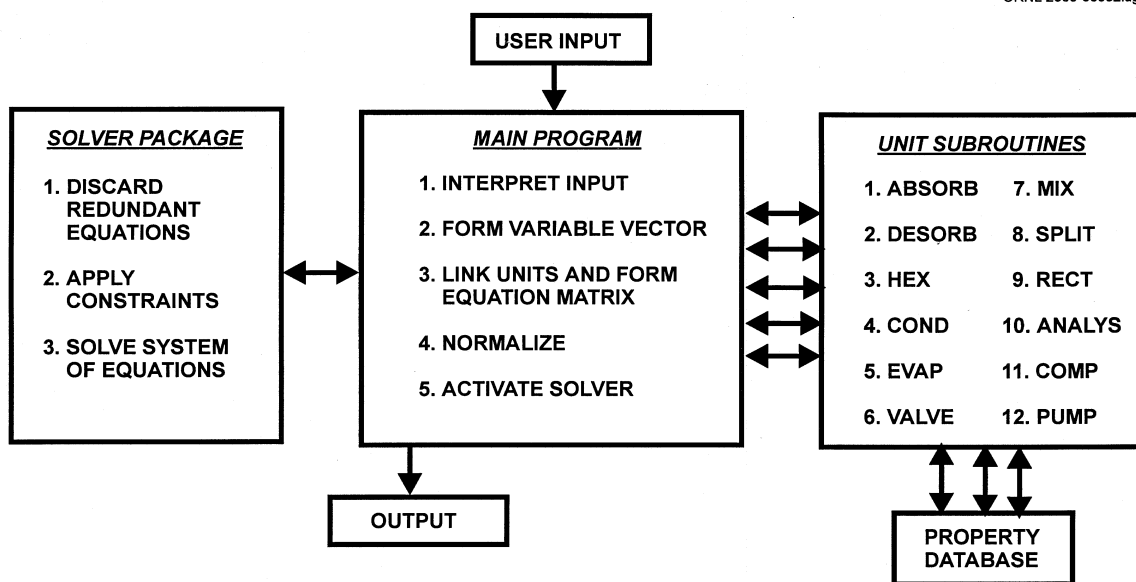


Fig. 1. Structure of the computer code.

### 3.1 THE INPUT

Unlike in most process simulators and flowsheeting programs commonly used by the chemical industry, the user of the present program does not have to do any programming nor write any of the equations. Furthermore, because all the governing equations are solved simultaneously, the user does not have to specify an iterative sequence by which the program is to proceed through the cycle, which eliminates the need to worry about the order of the variables and parameters. To prepare an input, the user first draws a schematic diagram depicting the cycle

to be simulated in terms of the basic 12 units recognizable by the code (Figure 1). A detailed description of these units and their governing equations is given in Sect. 4. The diagram may be drawn manually, on a sheet of paper, or directly on the computer screen using the ABSIM graphical interface. In the case of manual drawing, the diagram must display schematically the system's breakdown into the basic units and their interconnections. The user must label each state point and each unit; the order may be arbitrary.

The input file (which may be typed in the case of manual drawing or prepared interactively using the graphical interface) consists of the following three parts:

**General Information** — containing a user-supplied problem title, scaling parameters for normalization, limits on the number of iterations, and convergence criteria. Default options are available for all the above. The code can operate either in SI or British units, as indicated by a proper flag. The total numbers of units and state points are specified.

**Unit Input** — showing, for each of the units in the cycle, the unit number in the diagram, its type and sub-type, heat and mass transfer characteristics (or other characteristics as relevant), and a list of its state point numbers in the schematic diagram in an order corresponding to that of the unit subroutine. Heat transfer characteristics of a unit may be specified by one of six methods. The user can supply either the *UA* (overall heat transfer coefficient times area), the *NTU* (number of transfer units), the *EFF* (effectiveness), the *CAT* (closest approach temperature) or the *LMTD* (logarithmic mean temperature difference), or the heat duty. Mass transfer characteristics of a unit are specified by a temperature deviation from equilibrium at the liquid outlet (*DEVL*) and/or the vapor outlet (*DEVG*).

**State Point Input** — showing, for each of the state points in the cycle, the state point number, a code for the working fluid in it, and five pairs of numbers for the temperature, flowrate, concentration, vapor pressure, and vapor fraction in that state point. Each pair of numbers consists of an integer index (to specify whether the property is fixed or a variable) and a numerical value for that property. The latter gives a fixed value or an initial guess for the quantity in question.

To summarize, by this form of input, the user has conveyed to the program the complete necessary information about the cycle without the need for any programming. In the unit input, the program is told which units are contained in the cycle, what are their heat and mass transfer characteristics, and how they are interconnected. In the state point input, the program is told which working fluids are used, where, and what are the fixed and variable parameters of the problem. Additional details have been included in the Input Manual (Appendix A). Several examples of input preparation are included in ref. [13]. When the graphical interface is used, the input file \*.DAT is created automatically from the drawn diagram and from the data entered interactively.



## 3.2 THE MAIN PROGRAM

The main program is the coordinator among the different modules shown in Fig. 1. The principal tasks performed by the main program are interpreting the input and constructing a mathematical model of the cycle. The latter task includes forming a vector of normalized variables, linking the units and forming the system of equations and constraints, testing for and eliminating redundant equations and, if necessary, forming the pattern of the Jacobian. Then the solver is activated. The main program calls the unit subroutines and the solver package, when required, as illustrated in Fig. 1. By scanning through the State Point Input, the main program is able to identify the variable quantities by their non-zero flags. The physical variables are normalized by division by the proper scaling parameters, taken from the General Information section of the Input. The variable vector can contain unknown temperatures, flowrates, concentrations, pressures, and vapor fractions. Other unknown properties such as enthalpy are treated as dependent variables and are calculated at each iteration from the current values of the properties in the variable vector. Calling the unit subroutines in the arrangement given in the Unit Input enables the main program to collect the equations associated with each, and to substitute in them the appropriate variables and fixed quantities as mandated by the unit's interconnections. The equations are counted and normalized, using the scaling parameters, and each is assigned a number. During the solution process, the residuals of the equations are calculated in successive iterations.

Before beginning to solve, the program displays (at the user's request) the total number of equations and unknowns. Usually, the former exceeds the latter, as calls to unit subroutines sometimes generate redundant equations originating from essentially identical mass balances at different units. The redundant equations are eliminated automatically by the program. When all the unknown variables have been computed to the desired accuracy, the main program converts them back from the dimensionless values to physical values.

## 3.3 THE OUTPUT

The output structure is modular, aimed at giving the user control over the extent of the information presented. A flag in the General Information part of the input (*MSGLVL*) allows the user to specify how frequently values of the variables and equation residuals at intermediate iterations should be printed. This allows the user to follow the convergence process. Alternatively, the user can ask the code to print only final results. The results are presented in the form of a table listing the temperature, enthalpy, flowrate, concentration, pressure, and vapor

fraction at each state point, followed by another table listing the heat duty and transfer characteristics of each unit.

When the graphical interface is used, the output file \*.OUT is created automatically. The results of the calculations in it may be viewed on the computer screen, superimposed on the cycle diagram.

Further discussion and elaboration on the parts of the code mentioned above and illustrated in Fig. 1 may be found in ref. [13], including samples of input and output for some relatively simple absorption cycles. The following sections will discuss the remaining three modules of the code illustrated in Fig. 1: the unit subroutines, the property database, and the solver.

#### 4. UNIT SUBROUTINES AND GOVERNING EQUATIONS

Figure 2 illustrates the 12 standard units recognizable by the code, with their respective state points. The units, with their module types given in parentheses, are absorber (1), desorber (2), heat exchanger (3), condenser (4), evaporator (5), valve (6), mixer (7), splitter (8), rectifier (9), analyser (10), compressor (11) and pump (12). These units were found sufficient to create most absorption cycles of interest. Each unit type may have several sub-types differing slightly from each other, which will be discussed specifically below. The identity code (*IDUNIT*) assigned in the program to each unit is an integer composed of its type and sub-type as follows:

$$IDUNIT = 10*(unit\ type) + (unit\ sub-type) . \quad (4.1)$$

Thus, for example, an absorber (type 1) with the vapor in counterflow to the solution (sub-type 3) is assigned the identity code *IDUNIT* = 13.

Each unit is treated as a control volume, with its own inputs and outputs, that can be connected to other components. The governing equations apply to the unit as a whole and are formed from some or all of the following physical laws:

- Conservation of total mass:

$$\sum_i F_i = 0 . \quad (4.2)$$

- Conservation of mass for each material species (absorbent/absorbate):

$$\sum_i (F_i C_i) = 0 . \quad (4.3)$$

- Energy balance:

$$\sum_i (F_i h_i) = 0 . \quad (4.4)$$

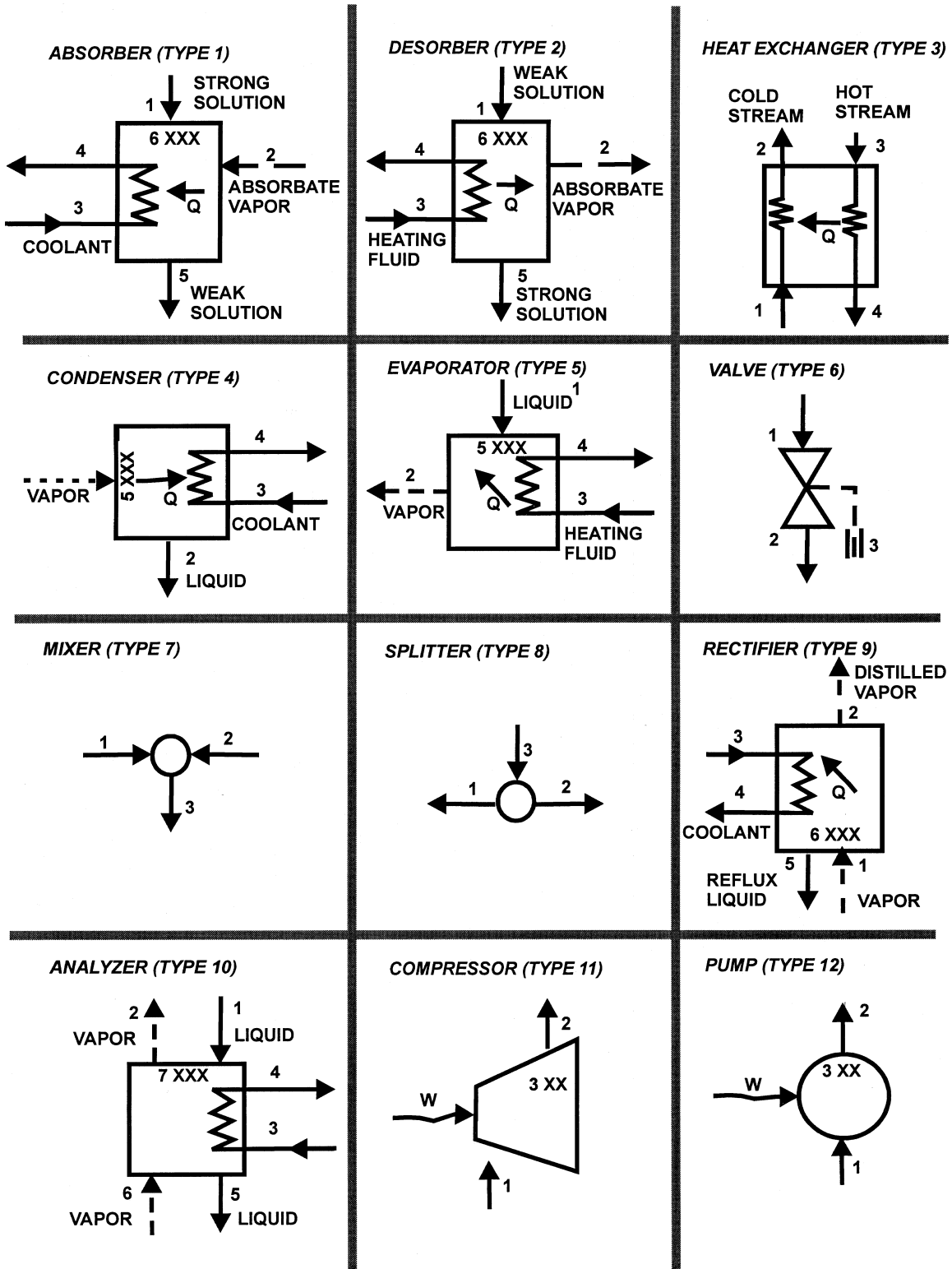


Fig. 2. Schematic description of individual units forming absorption systems.

- Heat transfer, expressed in one of the following six forms:

$$Q_{unit} - Q = 0, \quad (4.5a)$$

$$Q_{unit} - UA * \underline{LMTD} = 0, \quad (4.5b)$$

$$\underline{NTU} - NTU = 0, \quad (4.5c)$$

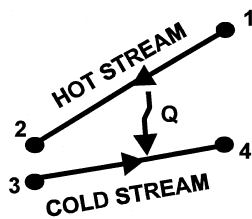
$$\underline{EFF} - EFF = 0, \quad (4.5d)$$

$$\underline{CAT} - CAT = 0, \text{ or} \quad (4.5e)$$

$$\underline{LMTD} - LMTD = 0, \quad (4.5f)$$

where  $Q$ ,  $NTU$ ,  $EFF$ ,  $CAT$ , and  $LMTD$  are user-supplied values of the heat duty, number of transfer units, heat transfer effectiveness, closest approach temperature, and logarithmic mean temperature difference, respectively; and  $Q_{unit}$ ,  $\underline{NTU}$ ,  $\underline{EFF}$ ,  $\underline{CAT}$ , and  $\underline{LMTD}$  are calculated values of these quantities in terms of the temperatures at each unit's state points. When specifying a unit in counterflow heat exchange in terms of  $EFF$  or  $CAT$ , the user must indicate the location of the temperature pinch, whether at the hot or cold end, by setting the integer flag  $IPINCH$  to +1 or -1, respectively. Figure 3 illustrates the two possibilities in counterflow heat exchange and defines  $CAT$  and  $EFF$  for each case. The definition of  $\underline{LMTD}$  is the same in both cases. By setting  $IPINCH = 0$ , the user lets the code decide the location of the pinch.

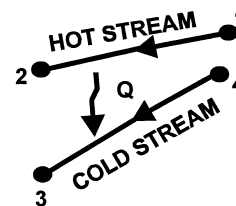
ORNL 2000-00034



PINCH AT COLD END,  $IPINCH = -1$

$$\underline{CAT} = T_2 - T_3$$

$$\underline{EFF} = (T_1 - T_2) / (T_1 - T_3)$$



PINCH AT HOT END,  $IPINCH = +1$

$$\underline{CAT} = T_1 - T_4$$

$$\underline{EFF} = (T_4 - T_3) / (T_1 - T_3)$$

$$\underline{LMTD} = \frac{(T_1 - T_4) - (T_2 - T_3)}{\ln [(T_1 - T_4) / (T_2 - T_3)]}$$

Fig. 3. The two possibilities for temperature variations in counterflow heat exchange.

- Vapor pressure–temperature–concentration (P-T-X) equilibrium between liquid and vapor:

$$f(P_i, T_i, C_i) = 0 \quad (4.6)$$

- Mass transfer, expressed in terms of temperature deviation from equilibrium (*DEV*):

$$T_i = T_{iE}(P_i, C_i) + DEV \quad (4.7)$$

(*DEV* is denoted by *DEVL* on the liquid side and by *DEVG* on the vapor side).

From these equations for each unit, a set of nonlinear equations is formed for the entire system and must be solved simultaneously. This step is done by expressing the equations in the form of functions, the residuals of which must be reduced by the solver to zero or, in practice, to a value below a given tolerance:

$$\begin{aligned} F_1(x_1, x_2, \dots, x_n) &= \epsilon_1, \\ F_2(x_1, x_2, \dots, x_n) &= \epsilon_2, \\ &\cdot \\ &\cdot \\ &\cdot \\ F_i(x_1, x_2, \dots, x_n) &= \epsilon_i, \\ &\cdot \\ &\cdot \\ F_n(x_1, x_2, \dots, x_n) &= \epsilon_n, \\ |\epsilon_i| &\rightarrow 0 \text{ at the solution} \end{aligned} \quad (4.8)$$

By its nature, a system of simultaneous nonlinear equations has many solutions. One of the main problems in developing the code has been to ensure that the solution is physically meaningful regardless of the initial guess. This goal was achieved by introducing constraints into the program. The constraints are of an inequality type and are generated by the unit subroutines when called, in much the same way as are the equations. Each unit has its own constraints that ensure, for example, that the heat flow is from a hot to a cold stream and not vice versa.

Following is a detailed listing, by unit, of the governing equations and constraints in each unit subroutine.

#### 4.1 ABSORBER (Subroutine ABSORB)

The absorber is referred to in the code as unit type 1 and is described schematically in Fig. 2. A strong liquid absorbent (state 1) enters the unit at a total pressure equal to or greater than the absorber pressure. It contacts absorbate vapor that may contain some liquid at state 2. Absorption occurs while heat is rejected to the coolant stream that enters at state 3 and leaves at state 4. The absorbent exits weaker at state 5. The entering, strong absorbent may be subcooled or superheated; it is therefore allowed to reach equilibrium at point 6 (point marked by XXX) either by adiabatic absorption or desorption [14] before the absorption process 6→5, accompanied by heat transfer, begins. The coolant may be any liquid (e.g., water or absorbent solution) or phase-changing pure substance (e.g., water evaporating into steam).

Three graphical configurations are available for the absorber with the vapor at state 2 in cross-flow, co-flow or counter-flow to the solution (sub-types 1, 2, and 3, respectively). The governing equations are

- overall mass balance,

$$F_1 + F_2 - F_5 = 0 ; \quad (4.1.1)$$

- absorbent/absorbate mass balance,

$$F_1 C_1 + F_2 C_2 - F_5 C_5 = 0 ; \quad (4.1.2)$$

- energy balance,

$$(F_1 h_1 + F_2 h_2 - F_5 h_5) - F_3 (h_4 - h_3) = 0 ; \quad (4.1.3)$$

- heat transfer [one of the equations (4.5a) – (4.5f)], where

$$Q_{unit} = F_3 (h_4 - h_3) , \quad (4.1.4a)$$

$$\underline{LMTD} = \frac{(T_6 - T_4) - (T_5 - T_3)}{\ln[(T_6 - T_4)/(T_5 - T_3)]} , \quad (4.1.4b)$$

$$\begin{aligned} \underline{CAT} &= (T_5 - T_3) \text{ for } IPINCH = -1 \text{ and} \\ &= (T_6 - T_4) \text{ for } IPINCH = +1 , \end{aligned} \quad (4.1.4c)$$

$$\begin{aligned} \underline{EFF} &= (T_6 - T_5)/(T_6 - T_3) \text{ for } IPINCH = -1 \text{ and} \\ &= (T_4 - T_3)/(T_6 - T_3) \text{ for } IPINCH = +1 ; \end{aligned} \quad (4.1.4d)$$

- mass transfer,

$$T_5 = T_{5E}(P_5, C_5) + DEVL ; \quad (4.1.5)$$

- adiabatic equilibrium at point 6 (if  $T_6 = T_1$ , then  $C_6 = C_1$  and  $F_6 = F_1$ ),

$$(h_1 - h_2)/(h_1 - h_6) = (C_1 - C_2)/(C_1 - C_6) , \quad (4.1.6)$$

$$T_6 = T_{6E}(P_6, C_6) , \quad (4.1.7)$$

$$F_6/F_1 = (C_1 - C_2)/(C_6 - C_2) . \quad (4.1.8)$$

Thus, we have a total of eight equations. The constraints are

$$\begin{aligned} T_6 &> T_4 , \\ T_5 &> T_3 , \\ T_4 &\geq T_3 . \end{aligned} \quad (4.1.9)$$

## 4.2 DESORBER (Subroutine DESORB)

The desorber is referred to in the code as unit type 2 and is described schematically in Fig. 2. A weak liquid absorbent (state 1) enters the unit at a total pressure equal to or greater than the desorber pressure. The absorbent receives heat from a heating fluid, which enters at state 3 and leaves at state 4, and also releases absorbate vapor at state 2. The absorbent exits stronger at state 5. The entering weak absorbent may be subcooled or superheated; it is therefore allowed to reach equilibrium at point 6 (point marked by XXX) either by adiabatic absorption or desorption [14] before the desorption process 6→5, accompanied by heat transfer, begins. The heat transfer fluid may be any liquid (e.g., water or absorbent solution) or phase-changing pure substance (e.g., condensing steam).

Three graphical configurations are available for the desorber with the vapor at state 2 in cross-flow, co-flow or counter-flow to the solution (sub-type 1, 2, and 3, respectively). The governing equations are

- overall mass balance,

$$F_2 + F_5 - F_1 = 0 ; \quad (4.2.1)$$



- absorbent/absorbate mass balance,

$$F_2C_2 + F_5C_5 - F_1C_1 = 0 ; \quad (4.2.2)$$

- energy balance,

$$(F_2h_2 + F_5h_5 - F_1h_1) - F_3(h_3 - h_4) = 0 ; \quad (4.2.3)$$

- heat transfer [one of the equations (4.5a) – (4.5f)], where

$$Q_{unit} = F_3(h_3 - h_4) , \quad (4.2.4a)$$

$$\underline{LMTD} = \frac{(T_4 - T_6) - (T_3 - T_5)}{\ln[(T_4 - T_6)/(T_3 - T_5)]} , \quad (4.2.4b)$$

$$\begin{aligned} \underline{CAT} &= (T_4 - T_6) \text{ for } IPINCH = -1 \text{ and} \\ &= (T_3 - T_5) \text{ for } IPINCH = +1 , \end{aligned} \quad (4.2.4c)$$

$$\begin{aligned} \underline{EFF} &= (T_3 - T_4)/(T_3 - T_6) \text{ for } IPINCH = -1 \text{ and} \\ &= (T_5 - T_6)/(T_3 - T_6) \text{ for } IPINCH = +1 ; \end{aligned} \quad (4.2.4d)$$

- mass transfer — liquid side,

$$T_5 = T_{5E}(P_5, C_5) + DEVL ; \quad (4.2.5)$$

- heat/mass transfer — vapor side,

$$\begin{aligned} T_2 &= T_5 + DEVG \text{ for sub-types 1, 2 and} \\ T_2 &= T_6 + DEVG \text{ for sub-type 3;} \end{aligned} \quad (4.2.6)$$

- equilibrium at point 2 (when  $C_2 \neq 0$ ),

$$C_2 = C_{2E}(T_2, P_2) ; \quad (4.2.7)$$

- adiabatic equilibrium at point 6 (if  $T_6 = T_1$  then  $C_6 = C_1$  and  $F_6 = F_1$ ),

$$(h_1 - h_2)/(h_1 - h_6) = (C_1 - C_2)/(C_1 - C_6) \quad (4.2.8)$$

$$T_6 = T_{6E}(P_6, C_6) , \quad (4.2.9)$$

$$F_6/F_1 = (C_1 - C_2)/(C_6 - C_2) . \quad (4.2.10)$$

Thus, we have a total of ten equations. The constraints are

$$\begin{aligned}
 T_3 &> T_5 , \\
 T_4 &> T_6 , \\
 T_5 &> T_6 , \\
 T_3 &\geq T_4 .
 \end{aligned}
 \tag{4.2.11}$$

### 4.3 HEAT EXCHANGER (Subroutine HEX)

The heat exchanger is referred to in the code as unit type 3 and is described schematically in Fig. 2. A hot stream entering at state 3 and leaving at state 4 transfers heat to a cold stream entering at state 1 and leaving at state 2. Either stream may comprise any liquid or gas not undergoing a phase change.

Three configurations are available for the heat exchanger with the hot stream  $3 \rightarrow 4$  in cross-flow, co-flow or counter-flow to the cold stream  $1 \rightarrow 2$  (sub-type 1, 2 and 3, respectively). The governing equations are

- energy balance,

$$F_3(h_3 - h_4) - F_1(h_2 - h_1) = 0 \quad \text{and} \tag{4.3.1}$$

- heat transfer [one of the equations (4.5a) – (4.5e)], where

$$Q_{unit} = F_1(h_2 - h_1) , \tag{4.3.2a}$$

$$LMTD = \frac{(T_3 - T_2) - (T_4 - T_1)}{\ln[(T_3 - T_2)/(T_4 - T_1)]} , \tag{4.3.2b}$$

$$\begin{aligned}
 \underline{CAT} &= (T_4 - T_1) \text{ for } IPINCH = -1 \text{ and} \\
 &= (T_3 - T_2) \text{ for } IPINCH = +1 ,
 \end{aligned}
 \tag{4.3.2c}$$

$$\begin{aligned}
 \underline{EFF} &= (T_3 - T_4)/(T_3 - T_1) \text{ for } IPINCH = -1 \text{ and} \\
 &= (T_2 - T_1)/(T_3 - T_1) \text{ for } IPINCH = +1
 \end{aligned}
 \tag{4.3.2d}$$

(for sub-type 3).

Thus, we have a total of two equations. The constraints are

$$\begin{aligned}
 T_3 &> T_2 \\
 T_4 &> T_1 \\
 T_3 &\geq T_4 \\
 T_2 &\geq T_1 .
 \end{aligned}
 \tag{4.3.3}$$

#### 4.4 CONDENSER (Subroutine COND)

The condenser is referred to in the code as unit type 4 and is described schematically in Fig. 2. A vapor that is either saturated or superheated at the condenser pressure enters the unit at state 1. If superheated, it first cools to the condensing temperature at state 5 (point marked by XXX). The condensation process takes place between states 5 and 2, and heat is rejected to a coolant, which may be any liquid or a phase-changing pure substance. The condensate leaving at state 2 is a saturated liquid, with a possible degree of subcool specified by the user.

Two graphical configurations are available for the condenser with the vapor at state 1 entering either from the side or from the top (sub-types 1 and 2, respectively), relative to the liquid condensate outlet (state 2) which is always at the bottom. The governing equations are

- energy balance,

$$F_3(h_4 - h_3) - F_1(h_1 - h_2) = 0 ; \tag{4.4.1}$$

- heat transfer [one of the equations (4.5a) – (4.5f)], where

$$Q_{unit} = F_3(h_4 - h_3) , \tag{4.4.2a}$$

$$\underline{LMTD} = \frac{(T_5 - T_4) - (T_2 - T_3)}{\ln[(T_5 - T_4)/(T_2 - T_3)]} , \tag{4.4.2b}$$

$$\begin{aligned}
 \underline{CAT} &= (T_2 - T_3) \text{ for } IPINCH = -1 \text{ and} \\
 &= (T_5 - T_4) \text{ for } IPINCH = +1 ,
 \end{aligned}
 \tag{4.4.2c}$$

$$\begin{aligned}
 \underline{EFF} &= (T_5 - T_2)/(T_5 - T_3) \text{ for } IPINCH = -1 \text{ and} \\
 &= (T_4 - T_3)/(T_5 - T_3) \text{ for } IPINCH = +1 ;
 \end{aligned}
 \tag{4.4.2d}$$

- mass transfer,

$$T_2 = T_{2E}(P_2, C_2) + DEVL ; \tag{4.4.3}$$

- equilibrium at point 5 (when  $C_5 \neq 0$  and  $T_1 \neq T_5$ ),

$$T_5 = T_{5E}(P_5, C_5) . \quad (4.4.4)$$

Thus, we have a total of four equations. The constraints are

$$\begin{aligned} T_5 &> T_4 , \\ T_2 &> T_3 , \\ T_4 &\geq T_3 , \\ T_1 &\geq T_5 , \\ T_5 &\geq T_2 . \end{aligned} \quad (4.4.5)$$

#### 4.5 EVAPORATOR (Subroutine EVAP)

The evaporator is referred to in the code as unit type 5 and is described schematically in Fig. 2. A subcooled liquid or saturated vapor-liquid mixture at the evaporator pressure enters the unit at state 1. If subcooled, it first heats to the evaporation temperature at state 5 (point marked by XXX). The evaporation process takes place between states 5 and 2, and heat is received from a heating fluid, which may be any liquid or phase-changing pure substance. The substance leaving at state 2 is a saturated vapor, with a possible degree of superheat specified by the user.

Two graphical configurations are available for the evaporator with the vapor at state 2 leaving either from the side or from the bottom (sub-types 1 and 2, respectively) relative to the liquid inlet (state 1) which is always at the top. The governing equations are

- energy balance,

$$F_3(h_3 - h_4) - F_1(h_2 - h_1) = 0 ; \quad (4.5.1)$$

- heat transfer [one of the equations (4.5a) – (4.5f)], where

$$Q_{unit} = F_3(h_3 - h_4) , \quad (4.5.2a)$$

$$\underline{LMTD} = \frac{(T_3 - T_2) - (T_4 - T_5)}{\ln[(T_3 - T_2) / (T_4 - T_5)]} , \quad (4.5.2b)$$

$$\underline{CAT} = (T_4 - T_5) \text{ for } IPINCH = -1 \text{ and}$$

$$= (T_3 - T_2) \text{ for } IPINCH = +1 , \quad (4.5.2c)$$

$$\begin{aligned} \underline{EFF} &= (T_3 - T_4)/(T_3 - T_5) \text{ for } IPINCH = -1 \text{ and} \\ &= (T_2 - T_5)/(T_3 - T_5) \text{ for } IPINCH = +1 ; \end{aligned} \quad (4.5.2d)$$

- equilibrium at point 5 (when  $T_5 \neq T_1$ ),

$$T_5 = T_{5E}(P_5, C_5) ; \quad (4.5.3)$$

- equilibrium at point 2 (when  $T_2 \neq T_5$ ),

$$C_2 = C_{2E}[P_2, (T_2 - DEVG)] . \quad (4.5.4)$$

Thus, we have a total of four equations. The constraints are

$$\begin{aligned} T_3 &> T_2 \\ T_4 &> T_5 , \\ T_3 &\geq T_4 , \\ T_2 &\geq T_5 , \\ T_5 &\geq T_1 . \end{aligned} \quad (4.5.5)$$

#### 4.6 VALVE (Subroutine VALVE)

The expansion valve is referred to in the code as unit type 6 and is described schematically in Fig. 2. Liquid or a liquid-vapor mixture enters at state 1 and expands to the lower pressure at state 2, yielding saturated vapor or liquid-vapor mixture.

Three configurations are available for the expansion valve. One is generic (sub-type 1), where the flowrate through the valve is specified by the user or by the rest of the cycle, regardless of the pressure difference across the valve. The second (sub-type 2) is a throttle valve with a given relation between the flowrate and the pressure difference. The third (sub-type 3) is a thermostatic expansion valve that uses the temperature sensor (state 3) to sense the temperature at any state point in the system, and controls the flowrate so as to maintain a fixed, user-specified temperature difference between state points 2 and 3. The governing equations are

- energy balance,

$$h_2 - h_1 = 0 ; \quad (4.6.1)$$

- equilibrium at point 2,

$$T_2 = T_{2E}(P_2, C_2) ; \quad (4.6.2)$$

- pressure difference-flowrate relation (for sub-type 2 only),

$$F_1 = C_{VLV}(P_1 - P_2)^p \quad (4.6.3)$$

where  $C_{VLV}$  is the valve coefficient and  $p$  is a power law coefficient;

- thermostatic operation (for sub-type 3 only),

$$T_3 - T_2 = \Delta T ; \quad (4.6.4)$$

where  $\Delta T$  is the temperature difference to be maintained as fixed.

#### 4.7 MIXER (Subroutine MIX)

The mixer is referred to in the code as unit type 7 and is described schematically in Fig. 2. Two streams (states 1 and 2) may be at different temperatures, flow rates, concentrations, and vapor fractions but enter the unit at the same pressure and combine to yield a third stream (state 3).

Two graphical configurations are available for the mixer with the two entering streams at 180° or at 90° to each other (sub-types 1 and 2, respectively). The governing equations are

- overall mass balance,

$$F_1 + F_2 - F_3 = 0 ; \quad (4.7.1)$$

- absorbent/absorbate mass balance,

$$F_1 C_1 + F_2 C_2 - F_3 C_3 = 0 ; \quad (4.7.2)$$

- energy balance,

$$F_1 h_1 + F_2 h_2 - F_3 h_3 = 0 ; \quad (4.7.3)$$

a total of three equations.

#### 4.8 SPLITTER (Subroutine SPLIT)

The splitter is referred to in the code as unit type 8 and is described schematically in Fig. 2. A stream at state 3 splits into two parts: 1 and 2.

Two graphical configurations are available for the splitter, with the two split streams at 180° or at 90° to each other (sub-types 1 and 2, respectively). The governing equations are

- overall mass balance,

$$F_3 - F_2 - F_1 = 0 ; \quad (4.8.1)$$

- split ratio (when specified),

$$F_1 = s F_3 ; \quad (4.8.2)$$

where  $s$  is the split ratio.

#### 4.9 RECTIFIER (Subroutine RECT)

The rectifier is referred to in the code as unit type 9 and is described schematically in Fig. 2. Saturated or superheated vapor at state 1 enters the unit at the rectifier pressure. If superheated, the vapor first cools down to equilibrium at state 6 (point marked by XXX). Then part of this vapor condenses as heat is rejected to the coolant and leaves the rectifier as saturated reflux liquid (state 5). The distilled vapor, now at saturation, leaves at state 2. The coolant may be any liquid or phase-changing pure substance. The governing equations are

- overall mass balance,

$$F_1 - F_2 - F_5 = 0 ; \quad (4.9.1)$$

- absorbent/absorbate mass balance,

$$F_1C_1 - F_2C_2 - F_5C_5 = 0 ; \quad (4.9.2)$$

- energy balance,

$$F_3(h_4 - h_3) - (F_1h_1 - F_2h_2 - F_5h_5) = 0 ; \quad (4.9.3)$$

- heat transfer [one of the equations (4.5a) – (4.5f)], where

$$Q_{unit} = F_3(h_4 - h_3) , \quad (4.9.4a)$$

$$LMTD = \frac{(T_6 - T_4) - (T_2 - T_3)}{\ln[(T_6 - T_4) / (T_2 - T_3)]} , \quad (4.9.4b)$$

$$\begin{aligned} CAT &= (T_2 - T_3) \text{ for } IPINCH = -1 \text{ and} \\ &= (T_6 - T_4) \text{ for } IPINCH = +1 , \end{aligned} \quad (4.9.4c)$$

$$\begin{aligned} EFF &= (T_6 - T_2) / (T_6 - T_3) \text{ for } IPINCH = -1 \text{ and} \\ &= (T_4 - T_3) / (T_6 - T_3) \text{ for } IPINCH = +1 ; \end{aligned} \quad (4.9.4d)$$

- mass transfer — liquid side,

$$T_5 = T_{5E}(P_5, C_5) + DEVL ; \quad (4.9.5)$$

- heat/mass transfer — vapor side,

$$T_6 = T_5 + DEVG ; \quad (4.9.6)$$

- equilibrium at point 2,

$$C_2 = C_{2E}(P_2, T_2) ; \quad (4.9.7)$$

- equilibrium at point 6 (when  $T_6 \neq T_1$ ),

$$C_6 = C_{6E}(P_6, T_6) . \quad (4.9.8)$$



Thus, we have a total of eight equations. The constraints are

$$\begin{aligned}
 T_6 &\geq T_5, \\
 T_6 &> T_4, \\
 T_5 &> T_4, \\
 T_4 &\geq T_3, \\
 T_2 &> T_3, \\
 T_6 &\geq T_2, \\
 T_1 &\geq T_6.
 \end{aligned}$$

#### 4.10 ANALYSER (Subroutine ANALYS)

The analyser is referred to in the code as unit type 10 and is described schematically in Fig. 2. A stream of liquid solution entering at state 1, possibly superheated or subcooled, reaches equilibrium at state 7 internally (point marked by XXX) and leaves at state 5. It interacts with a stream of vapor entering at state 6 and leaving at 2, in counterflow to the liquid. Heat may be added or removed through the stream 3 and 4.

Three configurations are available for the analyser. It may be externally heated, cooled, or operated adiabatically (sub-types 1, 2, and 3, respectively). The governing equations are

- overall mass balance,

$$F_1 + F_6 - F_2 - F_5 = 0 ; \quad (4.10.1)$$

- absorbent/absorbate mass balance,

$$F_1 C_1 + F_6 C_6 - F_2 C_2 - F_5 C_5 = 0 ; \quad (4.10.2)$$

- energy balance,

$$(F_1 h_1 + F_6 h_6 - F_2 h_2 - F_5 h_5) - F_3 (h_4 - h_3) = 0 ; \quad (4.10.3)$$

- heat transfer (for sub-types 1 and 2) [one of the equations (4.5a) – (4.5f)], where

$$\begin{aligned}
 Q_{unit} &= F_3 (h_3 - h_4) \text{ for sub-type 1 and} \\
 &= F_3 (h_4 - h_3) \text{ for sub-type 2;}
 \end{aligned} \quad (4.10.4a)$$

$$\underline{LMTD} = \frac{(T_4 - T_7) - (T_3 - T_5)}{\ln[(T_4 - T_7) / (T_3 - T_5)]}; \quad (4.10.4b)$$

$$\begin{aligned} \underline{CAT} &= (T_4 - T_7) \text{ for } IPINCH = -1 \text{ and} \\ &= (T_3 - T_5) \text{ for } IPINCH = +1, \text{ for sub-type 1;} \\ &= (T_5 - T_3) \text{ for } IPINCH = -1 \text{ and} \\ &= (T_7 - T_4) \text{ for } IPINCH = +1, \text{ for sub-type 2;} \end{aligned} \quad (4.10.4c)$$

$$\begin{aligned} \underline{EFF} &= (T_3 - T_4) / (T_3 - T_7) \text{ for } IPINCH = -1 \text{ and} \\ &= (T_5 - T_7) / (T_3 - T_7) \text{ for } IPINCH = +1, \text{ for sub-type 1;} \\ &= (T_7 - T_5) / (T_7 - T_3) \text{ for } IPINCH = -1 \text{ and} \\ &= (T_4 - T_3) / (T_7 - T_3) \text{ for } IPINCH = +1, \text{ for sub-type 2;} \end{aligned} \quad (4.10.4d)$$

- mass transfer — liquid side,

$$T_5 = T_{5E}(P_5, C_5) + DEVL; \quad (4.10.5)$$

- heat/mass transfer — vapor side,

$$T_2 = T_7 + DEVG; \quad (4.10.6)$$

- equilibrium at point 2,

$$C_2 = C_{2E}(T_2, P_2); \quad (4.10.7)$$

- adiabatic equilibrium at point 7 (if  $T_7 = T_1$  then  $C_7 = C_1$  and  $F_7 = F_1$ ),

$$(h_1 - h_2) / (h_1 - h_7) = (C_1 - C_2) / (C_1 - C_7) \quad (4.10.8)$$

$$T_7 = T_{7E}(P_7, C_7), \text{ and} \quad (4.10.9)$$

$$F_7 / F_1 = (C_1 - C_2) / (C_7 - C_2). \quad (4.10.10)$$

Thus, we have a total of ten equations. The constraints are

$$\begin{aligned}
 T_3 &\geq T_4, \\
 T_4 &> T_7, \\
 T_3 &> T_5, \\
 T_5 &> T_7 \\
 &\text{for sub-type 1 and} \\
 T_4 &\geq T_3 \\
 T_7 &> T_4 \\
 T_5 &> T_3 \\
 T_7 &> T_5
 \end{aligned} \tag{4.10.11}$$

for sub-type 2.

#### 4.11 COMPRESSOR (Subroutine COMP)

The compressor is referred to in the code as unit type 11 and is described schematically in Fig. 2. A stream of gas at low pressure enters at state 1 and is compressed to a high pressure (state 2). State 3 (marked by XXX) indicates the condition the gas would attain under isentropic compression. The governing equations are

- isentropic process,

$$S_3 - S_1 = 0 ; \tag{4.11.1}$$

- deviation from the ideal process,

$$(h_3 - h_1) - \eta(h_2 - h_1) = 0 ; \tag{4.11.2}$$

where  $\eta$  is the isentropic efficiency.

#### 4.12 PUMP (subroutine PUMP)

The pump is referred to in the code as unit type 12 and is described schematically in Fig. 2. A stream of liquid at low pressure enters at state 1, and is pumped to a high pressure (state 2).

State 3 (marked by XXX) indicates the condition the liquid would attain under isentropic pumping. The governing equations are

- isentropic process,

$$(h_3 - h_1) - (P_3 - P_1)/\rho_1 = 0 , \quad (4.12.1)$$

where  $\rho_1$  is the liquid density;

- deviation from the ideal process,

$$(h_3 - h_1) - \eta(h_2 - h_1) = 0 , \quad (4.12.2)$$

where  $\eta$  is the pumping efficiency.

## 5. MATERIAL PROPERTIES

As mentioned earlier, the thermodynamic properties of working fluids have been incorporated into the code in the form of a property database external to the unit subroutines. The purpose of doing so was twofold: first, to enhance modularity and flexibility by allowing different parts of a system to operate with different working fluids; and second, to allow for the extension of the database in the future and the addition of properties of other materials without affecting the rest of the code. Under this approach, each unit subroutine, when invoked, calls the database several times and retrieves from it the properties required by its various equations. Also, the material may vary from one state point to another, as specified by the user.

The structure of the database is illustrated in Fig. 4. Two main subroutines, EQB and ENTHAL, form the interface between the property database and the rest of the program and channel the flow of data to and from the specific fluid subroutines EQBi, one for each material  $i$ . Each of those, in turn, invokes additional routines, which produce information for the material  $i$  on vapor-liquid equilibrium relations among pressure, temperature, and liquid and vapor concentrations; on liquid and vapor enthalpies; and, in some cases, on liquid densities and vapor entropies. Subroutine EQB deals with the independent properties — temperatures, pressures and concentrations — and is called by the unit subroutines as they generate the equations at each iteration. Subroutine ENTHAL deals with the dependent property — enthalpy — and is called at the end of an iteration, after the independent properties have been determined at all state points for that iteration.

The structure of each specific fluid routine varies according to the nature of the substance and data available. Certain properties for some materials are easily described by a single formula over the entire applicable range; others require a very complicated and cumbersome formula to cover only part of the range, and the round-off error from substitution in such a formula is prohibitively large. In this case, tabular presentation of the properties with a suitable interpolation routine may be preferred. Furthermore, some materials behave quite differently in one part of the range than in other parts. Mixtures with a volatile absorbent such as ammonia-water exhibit very steep derivatives of temperature and pressure with respect to vapor concentrations at high compositions of the more volatile component. Therefore, whatever form was judged most convenient and accurate was used for each material in each specific fluid routine.

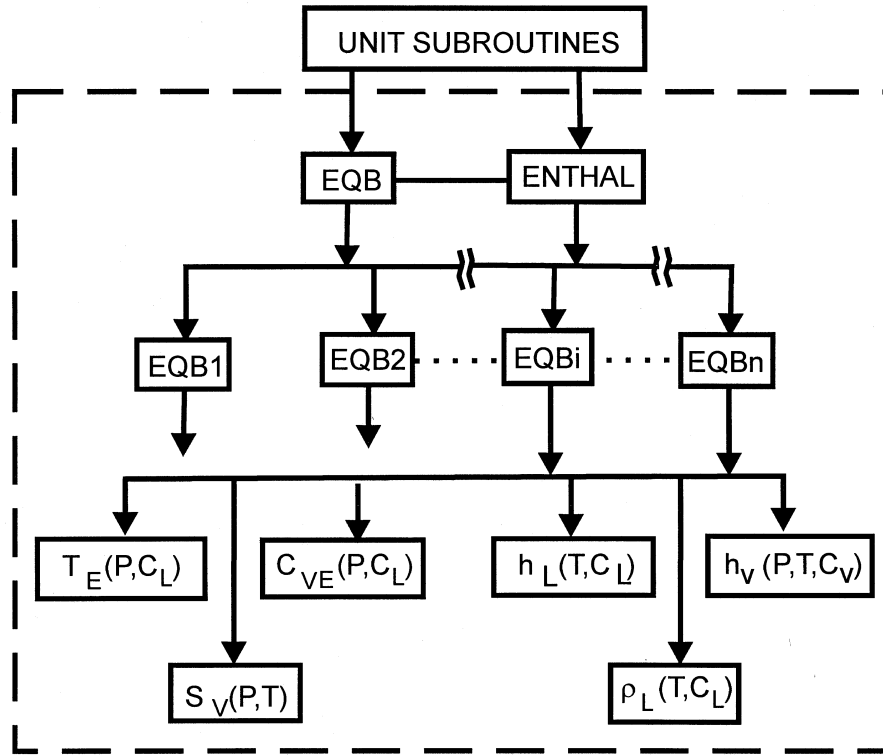


Fig. 4. Structure of the property database

The thermodynamic properties of working fluids have been extracted from the literature, with references given in [13]. The property database presently contains the following materials:

Material code	Material	Substance whose wt % concentration is specified
1	LiBr-H <sub>2</sub> O solution	LiBr
2	H <sub>2</sub> O-NH <sub>3</sub> solution and vapor	NH <sub>3</sub>
3	H <sub>2</sub> O liquid and vapor	(N/A)
4	(60%LiBr/40%H <sub>2</sub> O) -NH <sub>3</sub> solution	NH <sub>3</sub>
5	(28%LiBr/72%ZnBr <sub>2</sub> ) -CH <sub>3</sub> OH solution	LiBr/ZnBr <sub>2</sub>
6	CH <sub>3</sub> OH liquid and vapor	(N/A)
7	(53%LiNO <sub>3</sub> /28%KNO <sub>3</sub> /19%NaNO <sub>3</sub> ) -H <sub>2</sub> O solution	LiNO <sub>3</sub> /KNO <sub>3</sub> /NaNO <sub>3</sub>
8	NaOH-H <sub>2</sub> O solution	NaOH
9	LiCl-H <sub>2</sub> O solution	LiCl
10	Moist air	H <sub>2</sub> O (per dry air)
11	Flue gas (from CH <sub>4</sub> ): air,CO <sub>2</sub> , H <sub>2</sub> O	air

## 6. METHOD OF SOLUTION

The process by which the physical behavior of the absorption system to be simulated is translated by the code into governing equations has been described earlier. The equations are expressed in the form of functions whose residuals at the solution are to diminish to zero. Some of the equations are highly nonlinear, and the set therefore can have more than one mathematical solution. Also, the code often generates redundant equations as a result of multiple mass balances from different units within a closed loop. In order to solve it, it is therefore necessary (1) to identify and eliminate redundant equations, (2) to apply constraints and define the feasible region for the physically valid solution, and (3) to apply algorithms for solving the constrained set of equations.

The first step toward the solution is a properly constructed mathematical model of the cycle. To this end, the unit subroutines in the code are used in two different modes: first, to retrieve and construct the system of equations describing the entire cycle and the constraints on the variables; and second, to calculate the residual values of the functions during the solution process for the physical variables within the feasible region.

After reading the input data, the main program begins building the mathematical model of the cycle. This process is accompanied by testing the correctness of the input data. First, scaled mathematical variables  $X = (X_1, X_2, \dots, X_n)$  are defined, corresponding to the physical quantities to be calculated. Then, by calling the unit subroutines, the set of nonlinear equations:

$$R(X) = 0 \quad (6.1)$$

is formed, where  $R = (R_1, R_2, \dots, R_m)$ . Also, physical constraints are imposed on the variables in the form of upper and lower bounds:

$$A_i > X_i > B_i \quad i = 1, n \quad (6.2)$$

and so-called diagonal constraints:

$$X_i > X_j \quad (6.3)$$

for certain  $i$  and  $j$ . (An example of the latter constraint is the requirement set by the Second Law of Thermodynamics for the temperature at point  $i$  to be higher than at point  $j$  if heat flows from point  $i$  to point  $j$ ).

For a correctly defined cycle, the system [Eq. 6.1] should be neither overdetermined nor underdetermined. In the automated process of constructing the system [Eq. 6.1], two types of redundant equations may occur: linearly dependent equations (resulting from total and specific mass balances which are interdependent because of the existence of closed loops in the cycle); and nonlinear duplicate equations (resulting from the same physical law [e.g., thermodynamic equilibrium, applied in two different units at their point of connection]). These redundant equations are eliminated explicitly from the system [Eq. 6.1].

An absorption cycle may be treated as an oriented graph or network in which units — nodes and connections — are directed edges. Our linear equations describe the stream distribution in the nodes of the graph, and the number of independent equations is equal to the rank of the incidence matrix of the graph. This matrix is formed, and a Gauss elimination procedure is applied to select and eliminate the linearly dependent equations. This procedure is used separately for the truly linear equations of total mass balance and then in a somewhat more sophisticated form for the ‘quasi’-linear equations of specific material mass balance.

For elimination of duplicate nonlinear equations, the following algorithm is used. At the feasible initial point  $X_0$ , the values of the functions  $R_i(X)$  and the Jacobian  $J(X_0)$  are calculated. If certain functions are found to have the same values, and the elements of the derivatives are also the same, these functions are duplicate. Once identified, redundant ones are eliminated from the system [Eq. 6.1].

As a result of the above two procedures we are able to obtain, for a correctly defined absorption cycle, the system of Eq. (6.1) with an equal number of equations and unknowns ( $m = n$ ). If this condition is not satisfied, a diagnostic criterion is obtained of improper cycle definition in the input data. Elimination of redundant equations supplies the solver with a well-defined system.

The physical nature of the problem causes the Jacobian of the system [Eq. 6.1] to be sparse, involving usually no more than 10% to 25% non-zero elements. During the solution process, the finite difference approximation of the Jacobian is calculated several times, which takes  $n$  evaluations of all the functions  $R_i(X)$ . Using the sparsity pattern of the Jacobian makes it possible to reduce significantly the number of function evaluations for building the approximation with respect to the filling of non-zero elements. Therefore, on the last step of constructing the mathematical model, the code builds the sparsity pattern of the Jacobian  $J(X)$ .

The problem of calculating the performance parameters of an absorption cycle has now been reduced to the solution of a well-defined system of nonlinear algebraic equations: Eq. (6.1) with the constraints Eq. (6.2) and Eq. (6.3). Four characteristic features of the problem are (1) that the number, type and order of the equations vary from one case to another, depending on the cycle and working fluids; (2) that some of the functions in the system are not defined explicitly; (3)



that the residuals are to be calculated only within the feasible region Eq. (6.2) and Eq. (6.3) by the unit subroutines; and (4) that for the calculation of some functions, it is necessary to call the property database, a task that is often time-consuming.

An extensive search of available software has revealed no algorithms for the solution of systems of nonlinear equations with constraints of the present type. Reference [15] emphasizes the need for development of such algorithms. Standard methods of constrained optimization do not accommodate the specific features of the present problem and algorithms of nonlinear least squares with linear constraints [16] presume an analytical calculation of the Jacobian, while for the present problem this is unacceptable. We therefore preferred to use an algorithm for solving simultaneously a system of nonlinear equations with numerical approximation of the Jacobian and adapt it to the constrained situation by projection on the feasible region. A hybrid algorithm for unconstrained systems, developed by Powell [17], was selected. This method combines quasi-Newton and gradient approaches. Such a combination makes it possible to avoid complications due to a singular Jacobian, and to reduce the strict requirements on the initial guess. At the same time, the hybrid algorithm retains the efficiency of a quasi-Newton method.

The hybrid algorithm is described in detail in [17]. The basic ideas are as follows: Successive iterations are performed where each step is constrained by a repeatedly adjusted value  $\Delta_k$  which defines a so-called ‘trust region.’ For the current iterate, the displacement  $\delta_k$  of  $X^k$  is selected according to the following logic. Two searching directions are considered: first, the Newton step which has the form

$$\delta_n = -J^{-1}(X^k) * R(X^k), \quad (6.4)$$

where  $J(X^k)$  is an approximation of the Jacobian at the  $X^k$ ; and second, the steepest descent step toward the minimum of the function

$$f(X) = \sum [R_i(X)]^2 \quad (6.5)$$

along the antigradient direction, which can be expressed as

$$g = -\beta^* J^T(X^k) * R(X^k) \quad (6.6)$$

for definite  $\beta > 0$ . If  $\|\delta_n\| < \Delta_k$ , the step  $\delta_k$  is just the classical Newton correction  $\delta_k = \delta_n$ . If the inequality does not hold, the step Eq. (6.6) is tested. If  $\|g\| > \Delta_k$ , then the step of the length  $\Delta_k$  along the antigradient is executed. However, if  $\|g\| < \Delta_k$ , the displacement has also the length  $\Delta_k$

and is a linear combination of the Newton and gradient steps. The actual displacement to the new point  $X^{k+1} = X^k + \delta_k$  is realized only if

$$f(X^{k+1}) < f(X^k). \quad (6.7)$$

The hybrid method uses a numerical approximation of the Jacobian  $J(X)$ , which is updated at each iteration and recalculated by finite differences when necessary. Special conditions for adjustment of  $\Delta_k$  and updating the approximation of the Jacobian ensure convergence and efficiency of the hybrid algorithm [17].

For the constrained problem, the current point  $X^k$  should be inside the feasible region. There are several different approaches for keeping the current point feasible. We ensure the feasibility by projection on the polyhedron defined by the constraints Eq. (6.2) and Eq. (6.3). Vector  $\delta_k$  is calculated as in the original hybrid method. The point  $X^k + \delta_k$  is tested for feasibility and, if certain constraints are violated, the orthogonal projection  $X_{Pr}$  is calculated. The next stage of iteration is to try to estimate  $f(X_{Pr})$ . The functions  $R_i(X_{Pr})$ ,  $i = 1, n$  are therefore calculated at this point. If the expected inequality  $f(X_{Pr}) < f(X^k)$  holds, then the iteration defines  $X^{k+1} = X_{Pr}$ . However, if this condition fails, we let  $X^{k+1} = X^k$  and reduce the step-length restrictor  $\Delta_k$ . Because of the logic of the hybrid method, if  $\Delta_k$  is small enough, the searching direction coincides with the antigradient direction for function  $f(X)$ . In this case, one can take advantage of the gradient projection algorithm [18], which converges to the minimum of  $f(X)$ . This convergence ensures a decrease of the function  $f(X)$  and provides a criterion of existence of nonzero minimum  $f(X)$  on the boundaries. In the latter case, the algorithm stops and recommends that the user repeat the calculation with another initial guess. The rules for adjusting the step-length restrictor  $\Delta_k$  and reevaluating the Jacobian by finite-difference approximation are chosen so as to reduce the amount of work spent on searching the exact minimum of  $f(X)$  on the boundaries.

The procedure of testing the current point for feasibility and projecting it on the feasible region is implemented in a separate subroutine. The existence of the diagonal constraints [Eq. (6.3)] makes this task nontrivial, but still simpler than in the case of common linear constraints. For orthogonal projection, a method of quadratic programming is used [18] which comprises successive solutions of the system of linear equations. For our case, these systems are very sparse; an algorithm has been selected [19] that can take advantage of this sparsity.

## 7. USING THE GRAPHICAL INTERFACE

The graphical user interface has been described briefly in Sect. 2. Early versions of ABSIM had required that the user draw the cycle diagram on a sheet of paper and type the input file (\*.DAT) into the computer in tabular form as explained in Sect. 3. The results of the calculations were obtained in an output file (\*.OUT), also in tabular form, and had to be copied manually onto the cycle diagram. Even though running the program in this manual form is still an available feature, the graphical interface makes most of the tasks considerably easier. It enables the user to draw the cycle diagram on the computer screen, enter the input data interactively, run the program, and view the results either in the form of a table or superimposed on the cycle diagram. Special utilities enable the user to plot the results and to produce a P-T-X diagram of the cycle.

The main menu, which appears when activating AbsimW, includes the following entries:

- FILE — to manipulate ABSIM file
- ADD — to add units to the cycle
- DATA — to specify the operating conditions of the cycle
- RESULTS — to view the results on the cycle diagram
- PLOT — to plot certain calculated parameters vs. others
- RENUMBER — to change the number assigned to a unit or to a state point
- UTILITIES — to shape the cycle diagram, change text size and form
- HELP — to receive general instructions on ABSIM

AbsimW further contains a TOOL BOX, shown to the right of the screen, to assist in drawing or modifying the drawing of a cycle.

The following is a detailed description of each of these functions.

### 7.1 FILE

FILE enables the user to manipulate ABSIM files. Three primary files are usually associated with each cycle: \*.ABS is a graphical description file containing a map of the cycle diagram; \*.OUT is an ASCII file containing input parameters in the ABSIM format; and \*.OUT is an ASCII file containing the results of the calculations in tabular form. In addition, the code sometimes generates two secondary files associated with the cycle: \*.PTX contains information required to draw a P-T-X diagram of the cycle; and \*.PLT contains information required to plot certain parameters vs others, as requested by the user under PLOT. Of these, only \*.DAT and \*.PLT contain information that may be of interest to the user.

The FILE menu includes the following entries:

- OPEN — opens an existing cycle. A list of the cycles in the subdirectory \DATA will appear when this entry is selected.
- VIEW — enables the user to view ASCII files generated by ABSIM. Of particular interest are data files \*.DAT, output files \*.OUT, and plot files \*.PLT, as well as diagnostic files containing possible error messages. These files are found in the subdirectory \DATA.
- RUN — enables the user to run the simulation program after the cycle had been drawn and specified under DATA. With these tasks incomplete or improperly done, attempting to RUN will produce the message “run error!”

When RUN is selected, the graphical interface creates two files based on the user input, in the subdirectory \DATA (alongside the already existing \*.ABS): (1) an ASCII file \*.DAT that contains the input parameters in the ABSIM format; and (2) a temporary file TMP.TMP containing the names of the input, output, and P-T-X information files, typically \*.DAT, \*.OUT, and \*.PTX. The latter two are produced next (also in the subdirectory \DATA) when the ABSIM program is executed. \*.OUT contains the results of the calculations, including a list of the equations generated and their residuals, upon termination. In the case of nonconvergence, these residuals may be examined to indicate where the problem might be. \*.PTX contains information required for plotting a P-T-X diagram for the cycle.

A box containing a RUN MESSAGE appears on the screen at the conclusion of the run to report to the user whether convergence has been reached, the type of convergence, and the number of iterations it took.

Error messages generated by ABSIM during a run are collected in temporary files TMP.ERR and RUN.ERR. Messages and warnings other than error messages are collected in a temporary file TMP.STD. These temporary files are produced in the subdirectory \DATA and may be viewed by selecting VIEW under FILE.

- NEW — creates a new cycle. The name *UTILITIES* will be assigned to the cycle until it is saved under a different name.
- SAVE — saves the current file under a current name.
- SAVE AS — saves the current file under a new name (specify only filename, the extension is added automatically).
- PRINT — prints the cycle diagram. The diagram may be printed with or without the parameter values superimposed on it, depending on the way it is displayed on the screen at the time PRINT is selected.

- **PRINTER SETUP** — sets the printer characteristics and the print configuration.
- **EXIT** — exits AbsimW.

## 7.2 ADD

ADD enables the user, while drawing the cycle on the computer screen, to add one of the following 12 units specified under this entry to the current cycle diagram: absorber (3 configurations), desorber (3 configurations), heat exchanger (3 configurations), condenser (2 configurations), evaporator (2 configurations), valve (3 configurations), mixer (2 configurations), splitter (2 configurations), rectifier (3 configurations), analyser (3 configurations), compressor (2 configurations), and pump.

## 7.3 DATA

DATA enables the user to specify the characteristic parameters of a cycle (drawn earlier using ADD and the TOOL BOX), such as the heat and mass transfer characteristics of exchange units; the working fluid(s) at each state point; and the given temperatures, flowrates, and so on, fixed at certain state points. The data parameters belong to three basic groups (GLOBAL PARAMETERS, UNIT PARAMETERS, and STATE POINT DATA) as described below. Two additional options (READ DATA and COPY RESULTS) are available under DATA, as described below.

The DATA menu includes the following entries:

- **GLOBAL PARAMETERS** — the following parameters should be specified:
  - *TITLE* — enter a one-line title (up to 75 characters) for the current simulation problem.
  - *NORMALIZATION PARAMETERS* — provide values for the maximum temperature, minimum temperature, maximum flowrate and maximum pressure expected in the cycle, for normalization purposed. The values of these parameters need not be exact, and they are supplied automatically by the code from the STATE POINT DATA group if the user chooses not to enter them. Thus this step may be skipped.
  - *CONTROL PARAMETERS* — specify the maximum number of iterations, the frequency at which intermediate results of iteration are printed into the output file (MSGVLV), the system of units for calculation (SI/BRITISH), and the convergence tolerance on the functions (FTOL) and on the variables (XTOL). Default values are available for all these parameters.

- UNIT PARAMETERS — specify the characteristics of the units composing the cycle, as applicable (e.g., heat and mass transfer characteristics of a condenser, pressure difference/flowrate characteristic of a valve, isentropic efficiency of a compressor, and the like). Use the mouse to indicate a unit, then respond to the requests for data as they appear in the dialog box. This process should be repeated for all the units composing the cycle.
- STATE POINT DATA — specify the working fluid at each state point and a given (fixed) value or an initial guess for the temperature, flowrate, concentration, pressure, and vapor fraction at each state point. The value is either fixed, or a variable that is determined by the values of ITFIX, IFFIX, ICFIX, IPFIX, AND IWFIX for the above quantities, respectively, according to the following rule:

I\_FIX = 0 indicates a fixed value of the quantity.

I\_FIX = 1 indicates a variable value of the quantity, bearing no relation to the value of this quantity at other state points.

I\_FIX > 1 indicates a variable value of the quantity, identical at all the state points having the same I\_FIX.

When selecting the SP DATA entry from the DATA menu and selecting a particular quantity sub-entry, current values of this quantity (initially all zeros) will appear at all state points on the cycle diagram. Use the mouse to indicate a state point and enter the desired value of the quantity. Some values at other state points will be automatically adjusted, accordingly. For example, if the fluid type and flowrate at the inlet to a condenser are specified, the outlet fluid type and flowrate must be the same.

On the diagram, user-specified values are displayed in a slightly different color from calculated values, to help distinguish between the two. A hard copy of the cycle diagram with or without the data may be printed (see PRINT under FILE). To view the diagram without the data, select HIDE.

The DATA input process is now complete, and all the information required to properly define the absorption cycle is now available to the simulation program. Care must be taken to avoid under- or over-specification of the cycle, as this will lead to non-convergence. It is now recommended to save the file (by selecting SAVE from the FILE menu). The user is now ready to run the program. At this point, when RUN is selected from the FILE menu, an ASCII file \*.DAT will be created that contains the input parameters in the ABSIM format. This data file may be used to execute ABSIM from the DOS environment, bypassing the graphical interface. To do this under DOS, copy the files \*.DAT and TMP.TMP into the same directory with ABSIM.EXE, then type ABSIM at the DOS prompt. The files \*.OUT, \*.PTX containing the results of the calculations will be created under the same directory.

Two additional entries are available under the DATA menu:

- **READ DATA** — enables the user to skip the data input process described above by supplying a data file \*.DAT in the ABSIM format, compatible with the cycle diagram \*.ABS. This data file, which may be typed manually according to the format in the user's manual without using the graphical interface, must bear the same name as the graphical description file (with the extension .DAT instead of .ABS) and must be placed in the subdirectory \DATA along with the graphical description file. When this is done, select READ DATA.  
When READ DATA is executed, a dialog box appears asking the user whether to impose the state point interrelations built into ABSIM to ensure compatibility. Thus, for example, if the fluid type and flowrate at the inlet to a condenser are specified, the outlet fluid type and flowrate should be the same. If the state point interrelations are not satisfied in the user-supplied data file, ABSIM can impose them and modify the input accordingly. The choice is given to the user whether or not to exercise this option.
- **COPY RESULTS** — enables the user to improve the initial guesses for the cycle, by substituting the results of a successful calculation for the values currently in the data file. Having prepared the input according to the procedure described above, the user may RUN the program, and if convergence is achieved, COPY RESULTS should be selected. The results of the calculation will automatically replace the initial guesses for all the state point variables (temperatures, flowrates, etc.) in the original input. Running the program again now will, under most cases, yield convergence in fewer iterations than before. The new input with the updated initial guesses may provide a better starting point for the next calculation.

## 7.4 RESULTS

RESULTS enables the user to view the results of the calculations superimposed on the cycle diagram. When selecting the proper entry from the RESULTS menu, the program displays the temperature, flowrate, concentration, pressure or vapor fraction at each state point in the cycle on the respective streams in the diagram; or the  $UA$  (overall heat transfer coefficient times area),  $NTU$  (number of transfer units),  $EFF$  (heat transfer effectiveness),  $CAT$  (closest approach temperature),  $LMTD$  (logarithmic mean temperature difference) or heat duty in each unit in the cycle. Global results, including the coefficient of performance (COP) and capacity, may also be displayed. In each case, the quantity being displayed and its units of measurement are shown in the top header, along with the name of the file. Note that for state point parameters such as temperature and flowrate, user-specified values are displayed in a slightly different color than calculated values, to help distinguish between the two. A hard copy of the cycle diagram with or

without the results may be printed (see PRINT under FILE). To view the diagram without the results, select HIDE.

Selecting the entry PTX under RESULTS displays a P-T-X diagram of the cycle, based on the calculated results. This diagram appears under a separate window and may be printed by clicking the control box and selecting PRINT.

In case of nonconvergence, selecting RESULTS may help the user determine the source of the problem. A red frame will show along the contour of each unit in the cycle whose governing equations were not satisfied to the required tolerance. Failure of the code to reduce the residuals of certain equations may indicate improper specification of parameters associated with these equations, non-physical conditions and the like. Once the problematic units have been pointed out, the user should refer to the complete listing of equations and their residuals by viewing the file \*.OUT.

## 7.5 PLOT

PLOT enables the user to plot certain calculated parameters vs others. Typically, after completing a simulation run at a particular set of input conditions, the user may want to observe the influence of varying one parameter on the behavior of another. The graphical interface allows the user to specify the desired plot, display it on the screen and print it.

The PLOT menu includes the following entries:

- **CREATE** — specify the desired plot. The user is first asked whether to COPY RESULTS (see this feature under DATA) between points, in order to improve the initial guess for each calculation and help convergence. Next, a dialog box appears in which the user must specify: (1) an input parameter to be varied from a minimum to a maximum value at a specified number intervals; this parameter may be associated with a unit (e.g., a heat duty) or with a state point (e.g., a temperature); (2) the corresponding parameters to be plotted on the horizontal and vertical axes. These parameters may again be associated with a unit or with a state point, or they can be global parameters such as COP or capacity. Based on this specification, the input parameter is varied by the code over the required range and the program is run for each value to produce the required data, which are then written into a file \*.PLT. This ASCII file may be copied and used in other application programs, spreadsheets, and so forth. While PLOT is in progress, the status is displayed in terms of the percentage of the runs completed.



- **DISPLAY** — displays the plot on the screen. The two parameters plotted against each other are indicated on the horizontal and vertical axes, and the input parameter is indicated on top with the minimum and maximum values and the number of intervals selected. Note that the plot displays all the points calculated during the CREATE stage as requested by the user, regardless of whether the respective run has converged. To distinguish the former from the latter, a small square is used as a symbol for each point where convergence has been achieved, and an X for a point where convergence was not achieved. The displayed plot diagram appears under a separate window and may be printed by clicking the control box and selecting PRINT.

## **7.6 RENUMBER**

RENUMBER enables the user to change the number assigned by the code to a unit or to a state point. Several options are available under this entry, as indicated by the menu. To renumber units, use the mouse to indicate two units and their numbers will be swapped. To renumber state points, either indicate two state points to swap their respective numbers or indicate one state point and enter the number you want to assign to it. In certain units, you may merge an internal state point (point marked by XXX) with an external one, assigning the same number to both, when their conditions are identical.

## **7.7 UTILITIES**

UTILITIES enables the user to shape the cycle diagram and to change text size and form. The menu includes the following entries:

- **SET VIEW** — makes the current form of the cycle diagram into Base View — the form of the diagram to return to from various zooming and moving operations by means of the HOME button in the TOOL BOX.
- **TEXT SIZE** — allows the user to change the size of text, such as figures displaying values of parameters on the cycle diagram, relative to the default size. This feature is most useful in large cycles zoomed out to fit inside the window on the screen, where the numbers would look quite small. Use the mouse to select this entry and type in the dialog box the scale factor by which text size is to be magnified.
- **DECIMAL POINT** — sets the number of figures after the decimal point.

## 7.8 HELP

This entry enables the user to receive general instructions on ABSIM. Specific HELP instructions are also available under the green hypertext items at the various sub-entries.

## 7.9 TOOL BOX

TOOL BOX contains several devices to assist the user in drawing or modifying the drawing of a cycle diagram. It enables the user to move units and text in the cycle, zoom in and out, rotate units and text, align and flip units, erase units and text, and connect and disconnect units. TOOL BOX contains the following features, indicated by icons displayed to the right of the screen:

- **ARROW** — enables the use to move units and state point text in the cycle. To move a unit or state point text from one location to another in the cycle diagram, indicate the item by the arrow, press the left mouse button and drag it to the desired location. Note that when a unit is moved, its connections to other units will move with it.
- **ZOOM IN/OUT** — places the magnifying glass in the center of the desired zooming area. Click on the left mouse button to zoom in; click on the right mouse button to zoom out. The point indicated by the magnifying glass will relocate to the center of the window.
- **HOME** — returns to the original form of the diagram, referred to as Base View. The current form may be made into Base View by selecting the sub-entry SET VIEW under UTILITIES.
- **PAN-XY** — moves the entire cycle diagram up, down, right, or left. Place the pan-XY in the center of the diagram and move the mouse while pressing the left button.
- **ROTATE** — rotates units or state point text by 90°. Indicate the desired item and click on the left mouse button.
- **FLIP HORIZONTALLY** — flips a unit horizontally about a vertical axis.
- **FLIP VERTICALLY** — flips a unit vertically about a horizontal axis.
- **ALIGN HORIZONTALLY** — aligns a unit along the same horizontal axis with another unit. Use the mouse to indicate first the unit to be aligned, then the unit to align to.
- **ALIGN VERTICALLY** — aligns a unit along the same vertical axis with another unit. Use the mouse to indicate first the unit to be aligned, then the unit to align to.
- **ERASE** — erases a unit, a connection, or a state point text from a cycle diagram. Use the mouse to indicate the item and click on the left mouse button to erase it. If you erase a unit that had been connected to other units, its connecting streams will be erased along with it.

- **CONNECT** — connects an outlet stream (or sensor) from one unit to an inlet stream of another unit. A choice of solid, dashed, dotted or hidden lines is available for the connection. Use the mouse to mark the outlet. A set of crosshairs will appear. Use the mouse or the arrow keys (or Shift-Arrow) to advance the cross hairs in the desired direction in order to form segments of the connection; click on the left mouse button or press **INSERT** when completing a segment, or press **DELETE** to erase the last segment. When you reach the destination inlet, the connection is complete. The two connected state points, originally assigned two different numbers, will now have the same number.

Note that a unit may not be connected to itself. Also, certain units have more than one inlet or outlet; to help you identify the right stream, vapor streams (e.g., the vapor inlet into the absorber) has been colored purple, but liquid streams (e.g., the solution inlet into the same absorber) are blue.

## 8. RESULTS OF SIMULATION

The purpose of this section is to describe some of the capabilities of the ABSIM code. Many cycles have by now been simulated, and a detailed discussion of all the results is outside the scope of this document. This section therefore attempts to give a general overview with the most essential results, and references are given to other reports discussing specific systems for further details.

Prior to using ABSIM extensively for analysis of various advanced cycles, several initial runs were made for validation against experimental data taken from working absorption systems. Unfortunately, good, documented data are quite scarce. The literature describes a number of absorption systems developed during the past two decades, but only limited information is given on their heat transfer characteristics, with test data confined to a narrow range around the design point. Experimental data for our validation of the program were selected from measurements on a LiBr-H<sub>2</sub>O heat transformer for upgrading waste heat, developed and tested extensively at Oak Ridge National Laboratory [20]. Seventy-three experimental runs were documented where the machine had operated under steady-state, stable conditions for 1 to 8 h. The results of the validation have been described in detail in ref. [12]. Very good agreement was obtained between the simulation and experimental results.

Earlier versions of the code have been used to simulate more than 30 various cycles of interest with the working fluids LiBr-H<sub>2</sub>O, H<sub>2</sub>O-NH<sub>3</sub>, LiBr-H<sub>2</sub>O-NH<sub>3</sub>, and LiBr/ZnBr<sub>2</sub>-CH<sub>3</sub>OH. These cycles included single-, double-, and triple-effect absorption chillers, heat pumps, and heat transformers in various configurations. Some results for typical cases showing design point performance have been described in ref. [13]. Most systems with a working fluid involving a nonvolatile absorbent usually converged without difficulty; problems were encountered concerning fluids with a volatile absorbent such as H<sub>2</sub>O-NH<sub>3</sub> because of the highly nonlinear behavior of these materials at large concentrations of the volatile component in the vapor phase.

A detailed parametric study using ABSIM has been conducted to investigate the performance of various cycles using LiBr-H<sub>2</sub>O [21]. Complete performance maps under varying operating conditions have been generated for systems in single-effect and several double-effect configurations, in series and in parallel connection. The lack of flexibility in system-specific simulation models may be the reason why, to date, only a few comparative studies of advanced cycles have been carried out [6,7,8]. In the published literature describing these studies, a cycle's performance is often characterized by no more than a single point. Here, the performance has been compared over the entire temperature domain applicable to the cycle. The temperature range suitable for transition from single- to double-effect systems has been identified. A comparison was conducted between double-effect systems in series and parallel flow, showing

the COP advantage of the latter. The optimum split of solution flowrate in the parallel system has been determined. Some of the cycles investigated under this study [21] have been provided as samples with the present version of ABSIM (subdirectory DATA) and are included in Appendix B. Figure B.1 is the cycle diagram for a single-effect LiBr-H<sub>2</sub>O chiller (SEC) as drawn on the screen and printed using the graphical interface. The corresponding Table B.1 is the input file SEC.DAT generated for this cycle, and Table B.2 is the output file SEC.OUT. Figure B.2 describes a double-effect LiBr-H<sub>2</sub>O chiller in series flow Type 1 [21] (DECS1). Figure B.3 describes a double-effect LiBr-H<sub>2</sub>O chiller in parallel flow (DECP).

The LiBr-H<sub>2</sub>O study [21] of single- and double-effect systems has been extended to triple-effect cycles [22]. Figure B.4 describes a triple-effect chiller in parallel connection (DCCAP) formed as an extension of the corresponding double-effect system (DECP) by maintaining the same heat transfer characteristics and adding one desorber, one condenser, and one heat exchanger of the same size as in the DECP. Cooling is produced in evaporator 1 fed by water condensate from three condensers, while high-temperature heat from an external source is supplied to desorber 13 only. Desorbers 3 and 4 are powered by rejected heat from condensers 6 and 14, respectively, as shown. The refrigerant flow configuration is slightly different here from the one in the DECP (Figure B.3) and uses double-condenser-coupling (DCC) [22] to recover heat from the hot condensate leaving the high-temperature condensers and add it to the lower-temperature desorbers. A considerably higher COP is obtained here, compared with the corresponding single- and double-effect systems.

One of the main absorption programs currently in the United States concerns the development of a residential heat pump based on the generator-absorber exchange (GAX) cycle [23]. When a working fluid with a wide solution field, such as ammonia-water, is used in a single-stage system with a high regeneration temperature, some temperature overlaps between the absorber and desorber occur, making it possible to recover heat from the former to the latter and obtain some 'free' regeneration. The possibility of using GAX in combination with a solution-cooled absorber and a solution-heated generator is described in detail in ref. [23] and has been contemplated for some time. Some simplified calculations were performed for COP estimation under Phase 1 of the DOE/Oak Ridge National Laboratory advanced cycles development program [23]. However, a detailed calculation has been a formidable task because of the complexity of the cycle, combined with the highly nonlinear nature of the working fluid NH<sub>3</sub>-H<sub>2</sub>O. The current version of ABSIM makes it possible to perform such detailed simulation. Figure B.5 describes the GAX heat pump in terms of the units recognizable by the code. The absorber and desorber are each split into three parts. Absorber 12 is externally cooled, absorber 11 is solution-cooled and absorber 5 is GAX-cooled; similarly, desorber 3 is externally heated, desorber 2 is solution-heated and desorber 13 is GAX-heated, as shown.

## 9. REFERENCES

1. G. C. Vliet, M. B. Lawson and R. A. Lithgow. "Water-Lithium Bromide Double Effect Absorption Cooling Cycle Analysis." *ASHRAE Transactions*, 88, Part 1, pp. 811–23 (1982).
2. G. Grossman, J. L. Lando, I. Vardi, J. R. Bourne, Y. Kimchi and J. Ben-Dror. "Solar Powered Environment Control Criteria and Realization." *Proceedings, the 1979 International Solar Energy Society Conference*, Atlanta, Ga., May 1979, K.W. Boer and B.H. Glenn, eds., 1, pp. 720–24, Pergamon Press, 1979.
3. G. Grossman and K. W. Childs. "Computer Simulation of a Lithium Bromide-Water Absorption Heat Pump for Temperature Boosting." *ASHRAE Transactions*, 89, Part 1b, pp. 240–48 (1983).
4. M. O. McLinden and S. A. Klein. "Steady State Modeling of Absorption Heat Pumps with a Comparison to Experiments." *ASHRAE Transactions*, 91, Part 2b, pp. 1793–1807 (1985).
5. H. Perez-Blanco and M. R. Patterson. *Conceptual Design and Optimization of a Versatile Absorption Heat Transformer*. ORNL/TM-9841, Oak Ridge National Laboratory (1986).
6. W. J. Biermann. *Advanced Absorption Heat Pump Cycles*. ORNL/TM/Sub/81-17499/1, Oak Ridge National Laboratory (1984).
7. B. A. Phillips. "Analysis of Advanced Residential Absorption Heat Pump Cycles." *Proceedings, the DOE/ORNL Heat Pump Conference*, Washington D.C., December 1984, pp. 265–231.
8. R. J. Modahl. *Development and Proof Testing of Advanced Absorption Refrigeration Cycle Concepts*. ORNL/Sub/86-17498/1, Oak Ridge National Laboratory (1992).
9. L. B. Evans. "Process Flowsheeting: a State of the Art Review." *Proceedings, CHEM COMP 1982: Chemical Process Analysis and Design Using Computers*, G. F. Forment, ed., 1-12, K. Vlaam Ingenieursver, Anttwerp, Belgium (1982).
10. P. W. Gallier, L. B. Evans, H. I. Britt, J. F. Boston and P. K. Gupta. *ASPEN—Advanced Capabilities for Modeling and Simulation of Industrial Processes*. ACS Symposium Series 124, pp. 293–308 (1980).
11. C. C. Pantelides. "SPEEDUP—Recent Advances in Process Simulation." *Computers in Chemical Engineering*, 12, pp. 745–55 (1988).
12. G. Grossman and E. Michelson. "A Modular Computer Simulation of Absorption Systems." *ASHRAE Transactions*, 91, Part 2b, pp. 1808–27 (1985). Also ORNL/Sub/83-43337/2, Oak Ridge National Laboratory (1986).

13. G. Grossman, K. Gommed and D. Gadoth. "A Computer Model for Simulation of Absorption Systems in Flexible and Modular Form." *ASHRAE Transactions*, 93, Part 2, pp. 2389–2428 (1987). Also ORNL/Sub/90-89673, Oak Ridge National Laboratory (1990).
14. G. Grossman. "Adiabatic Absorption and Desorption for Improvement of Temperature-Boosting Absorption Heat Pumps." *ASHRAE Transactions*, 88, part 2, pp. 359–67 (1982).
15. K. L. Hiebert. "An Evaluation of Mathematical Software That Solves Systems of Nonlinear Equations." *Association for Computing Machinery Transactions on Mathematical Software (ACM TOMS)*, 8, pp. 5–20 (1982).
16. S. J. Wright and J. N. Holt. "Algorithms For Nonlinear Least Squares With Linear Inequality Constraints." *SIAM J. Scientific and Statistical Computation* 6, No. 4, pp. 1033–48 (1985).
17. M. J. D. Powell. "A Hybrid Method for Nonlinear Equations." In *Numerical Methods for Nonlinear Algebraic Equations*, P. Rabinowitz, ed., Gordon & Breach, London, 1970, pp. 87–114.
18. B. T. Polyak. *Introduction to Optimization (in Russian)*. Nauka, Moscow, pp. 296–300 (1983).
19. J. K. Reid. *Two Fortran Subroutines for Direct Solution of Linear Equations Whose Matrix is Sparse, Symmetric and Positive-definite*. Atomic Energy Research Establishment (AERE) Report R-7119, (1971).
20. W. R. Huntley. "Performance Test Results of an Absorption Heat Pump Using Low-Temperature (60°C) Industrial Waste Heat." *Proceedings, the 18th Intersociety Energy Conversion Engineering Conference, (IECEC)*, Orlando, Florida, August 1983, 4, pp. 1921–26, (1983). Also ORNL TM-9072, Oak Ridge National Laboratory (1984).
21. K. Gommed and G. Grossman. "Performance Analysis of Staged Absorption Heat Pumps: Water—Lithium Bromide Systems." *ASHRAE Transactions*, 96, Part 1, pp. 1590–98, (1990).
22. G. Grossman, M. Wilk and R. C. DeVault. "Simulation and Performance Analysis of Triple-Effect Absorption Cycles." *ASHRAE Transactions*, 100, Part 1, pp. 452–62, (1994).
23. B. A. Phillips: "Development of a Gas-Fired Heat Pump with an Improved Absorption Cycle." *Proceedings, the 1988 ASME Winter Annual Meeting*, Chicago, Illinois, November 27–December 2, 1988: Analysis and Applications of Heat Pumps, Vol. AES-8/SED-6, pp. 97–102.

## NOMENCLATURE

<i>A</i>	=	heat transfer area ( $m^2$ or $ft^2$ )
<i>C</i>	=	absorbent/absorbate concentration (wt %)
<i>CAT</i>	=	closest approach temperature ( $^{\circ}C$ or $^{\circ}F$ )
<i>DEV</i>	=	extent of deviation from equilibrium state ( $^{\circ}C$ or $^{\circ}F$ )
<i>DEVG</i>	=	deviation from equilibrium, vapor side ( $^{\circ}C$ or $^{\circ}F$ )
<i>DEVL</i>	=	deviation from equilibrium, liquid side ( $^{\circ}C$ or $^{\circ}F$ )
<i>EFF</i>	=	heat transfer effectiveness (dimensionless)
<i>F</i>	=	mass flow rate (kg/s or lb/min)
<i>f</i>	=	sum of squares of the functions, Eq.(6.5) (dimensionless)
<i>g</i>	=	steepest-descent step, Eq. (6.6) (dimensionless)
<i>h</i>	=	specific enthalpy (kJ/kg or Btu/lb)
<i>J(X)</i>	=	Jacobian matrix of the system of equations
<i>LMTD</i>	=	logarithmic mean temperature difference ( $^{\circ}C$ or $^{\circ}F$ )
<i>m</i>	=	number of equations
<i>n</i>	=	number of unknowns
<i>P</i>	=	pressure ( kPa or psia)
<i>Q</i>	=	heat transfer rate (kW or Btu/min)
<i>R</i>	=	symbol for function, Eq. (6.1) (dimensionless)
<i>S</i>	=	entropy (kJ/kg K or Btu/lb $^{\circ}R$ )
<i>T</i>	=	temperature ( $^{\circ}C$ or $^{\circ}F$ )
<i>U</i>	=	overall heat transfer coefficient (kW/ $m^2 \cdot ^{\circ}C$ or Btu/ $ft^2 \cdot ^{\circ}F \cdot min$ )
<i>UA</i>	=	overall heat transfer coefficient times area (kW/ $^{\circ}C$ or Btu/ $^{\circ}F \cdot min$ )
<i>W</i>	=	vapor mass fraction (dimensionless)
<i>X</i>	=	symbol for unknown variable (dimensionless)
$\Delta$	=	step-length restrictor, defining “rust region” (dimensionless)
$\delta_k$	=	current step (dimensionless)
$\delta_n$	=	Newton step, Eq. (18)
$\rho$	=	density (kg/ $m^3$ or lb/ $ft^3$ )

### Subscripts

<i>E</i>	=	equilibrium
<i>i, j</i>	=	state point index (numerical values may be used)
<i>k</i>	=	number of current iteration
<i>L</i>	=	liquid
<i>Pr</i>	=	projection
<i>V</i>	=	vapor



## APPENDIX A: INPUT MANUAL

Section 3 has described the structure of the input to the ABSIM code. This manual describes in detail the parameters required on each card of the input, their mathematical symbols, and their formats.

### A.1. GENERAL INFORMATION

The first card allows the user to supply a problem title. If no title is desired, input 'blanks.' The card has the following format:

Title card	Columns	1	<i>2-76</i> <i>ATITLE</i>
	Format	1X	<i>A75</i>
Explanation			
<i>ATITLE</i>		Name or title of problem, may consist of any characters	

The second card includes scaling parameters for normalization and has the following format:

Scaling parameter card	Columns	1-10 <i>TMAX</i>	11-20 <i>TMIN</i>	21-30 <i>FMAX</i>	31-40 <i>PMAX</i>
	Format	F10.1	F10.1	F10.1	F10.1
Explanation					
<i>TMAX</i>		Maximum expected temperature in the system			
<i>TMIN</i>		Minimum expected temperature in the system			
<i>FMAX</i>		Maximum expected mass flowrate in the system			
<i>PMAX</i>		Maximum expected pressure in the system			

The third card contains control parameters and has the following format:

Control parameter card	Columns	1-5 <i>MAXFEV</i>	6-10 <i>MSGLVL</i>	11-15 <i>IUFLAG</i>	16-25 <i>FTOL</i>	26-35 <i>XTOL</i>
	Format	I5	I5	I5	D10.1	D10.1
Explanation						
<i>MAXFEV</i>	Maximum number of iterations allowed before termination.					
<i>MSGLVL</i>	Integer flag controlling the printing of intermediate results					
= 0	No intermediate results are printed.					
= n	Intermediate results are printed every n iterations					
<i>IUFLAG</i>	Integer flag indicating the system of units used for the input and final output results (intermediate results are always in British units)					
= 1	Input and output in British units					
= 2	Input in British units; output in SI units					
= 3	Input and output in SI units					
= 4	Input in SI units; output in British units					
<i>FTOL</i> > 0	Convergence tolerance on the value of the functions' residuals. Exit from the solver routine occurs when the Euclidean norm of the residuals is less than <i>FTOL</i> .					
<i>XTOL</i> > 0	Convergence tolerance on the values of the variables. Exit from the solver routine occurs when the relative error between two successive iterations is at most <i>XTOL</i> .					

The fourth card indicates the number of units and state points in the system and has the following format:

System card	Columns	1 - 4 <i>NUNITS</i>	5 - 8 <i>NSP</i>
	Format	I4	I4
Explanation			
<i>NUNITS</i>		Total number of units or components	
<i>NSP</i>		Total number of state points	

## A.2. UNIT INPUT

This part of the input describes the units that make up the system. The user must supply two data cards for each unit. The first card provides a unit description and has the following format:

Unit I.D. card	Columns	1-5 <i>NU</i>	6-10 <i>IDUNIT</i>	11-15 <i>IHT</i>	16-26 <i>HT</i>	27-31 <i>IPINCH</i>	32-42 <i>DEVL</i>	43-53 <i>DEVG</i>	54-58 <i>ICOP</i>
	Format	I5	I5	I5	D11.4	I5	D11.4	D11.4	I5
Explanation									
	<i>NU</i>	Number assigned to the unit in the cycle diagram							
	<i>IDUNIT</i>	Unit ID (composed of unit type and sub-type)							
	= 1(3)	Absorber (cross-flow, co-flow, counter-flow)							
	= 2(3)	Desorber (cross-flow, co-flow, counter-flow)							
	= 3(3)	Heat Exchanger (cross-flow, co-flow, counter-flow)							
	= 4(2)	Condenser (side, top)							
	= 5(2)	Evaporator (side, bottom)							
	= 6(3)	Valve (generic, throttle, thermostatic)							
	= 7(2)	Mixer (inlet streams at 180°, 90°)							
	= 8(2)	Splitter (outlet streams at 180°, 90°)							
	= 9(3)	Rectifier							
	=10(3)	Analyser (heated, cooled, adiabatic)							
	=11(2)	Compressor							
	=12(1)	Pump							

<i>IHT</i>	Integer index indicating method of specification of unit heat transfer characteristics
= 0	Heat specified (BTU/min or kW)
= 1	UA Method (BTU/min- °F or kW/ °C)
= 2	NTU Method
= 3	EFF Method
= 4	CAT Method (°F or °C)
= 5	LMTD Method (°F or °C)
= 6	Skip
<i>HT</i>	Value of <i>Q</i> , <i>UA</i> , <i>EFF</i> , <i>NTU</i> , <i>CAT</i> or <i>LMTD</i> according to <i>IHT</i> index
<i>IPINCH</i>	Integer index indicating location of temperature pinch
= -1	Pinch at cold end
= +1	Pinch at hot end
= 0	Program decides the location of the pinch
<i>DEVL</i>	Temperature deviation from equilibrium on liquid side
<i>DEVG</i>	Temperature deviation from equilibrium on vapor side
<i>ICOP</i>	Integer index controlling calculation of COP
= 1	Heat duty added to COP numerator
= -1	Heat duty added to COP denominator
= 0	Heat duty not included in COP calculations

The second card describes how the unit is connected to other units in the system. The user must list the numbers of the state points associated with the unit in the system in an order corresponding to that in the unit subroutine. The card has the following format:

State Point Order card	Columns	1-4 ISP-1	5-8 ISP-2	9-12 ISP-3	13-16 ISP-4	17-20 ISP-5	21-24 ISP-6	25-28 ISP-7
	Format	I4	I4	I4	I4	I4	I4	I4
	Explanation							
	<i>ISP</i>	State point number to be listed in the order 1 to 7, corresponding with the unit subroutine						

### A.3. STATE POINT INPUT

This part of the input requires one card for each state point in the cycle. The card gives the state point number, a code for the working fluid in it, and five pairs of numbers for the temperature, mass flowrate, concentration, pressure, and vapor fraction in that state point. In each pair of numbers, the first is an integer indicating a fixed or variable quantity, and the second is a real number giving a fixed value or an initial guess for the quantity in question. The user must supply for each state point one input card that has the following format:

Initial Values card	Columns	1-4 <i>NDUM</i>	5-8 <i>KSUB</i>	9-11 <i>ITFIX</i>	12-22 <i>T</i>	23-25 <i>IFFIX</i>	26-36 <i>F</i>
	Format	I4	I4	I3	D11.4	I3	D11.4
	Columns	37-39 <i>ICFIX</i>	40-50 <i>C</i>	51-53 <i>IPFIX</i>	54-64 <i>P</i>	65-67 <i>IWFIX</i>	68-78 <i>W</i>
	Format	I3	D11.4	I3	D11.4	I3	D11.4
Explanation							
<i>NDUM</i>	State point number in the cycle diagram						
<i>KSUB</i>	type of working fluid						
= 1	LiBr-H <sub>2</sub> O		Solution				
= 2	H <sub>2</sub> O-NH <sub>3</sub>		Solution and vapor				
= 3	H <sub>2</sub> O		Liquid and vapor				
= 4	LiBr-H <sub>2</sub> O-NH <sub>3</sub>		Solution				
= 5	LiBr/ZnBr <sub>2</sub> -CH <sub>3</sub> OH		Solution				
= 6	CH <sub>3</sub> OH		Liquid and vapor				
= 7	LiNO <sub>3</sub> /KNO <sub>3</sub> /NaNO <sub>3</sub> -H <sub>2</sub> O		Solution				
= 8	NaOH - H <sub>2</sub> O		Solution				
= 9	LiCl - H <sub>2</sub> O		Solution				
= 10	Moist Air						
= 11	Flue Gas						
<i>ITFIX</i>	Integer index for temperature						
= 0	Fixed temperature						
= 1	Variable temperature, not equal to the temperature at any other state point						
≥ 2	Variable temperature. State points for which this integer is the same have identical temperatures.						
<i>T</i>	Initial value of temperature						

IFFIX	Same as ITFIX, for mass flowrate instead of temperature
F	Initial value of mass flow rate
ICFIX	Same as ITFIX, for concentration instead of temperature
C	Initial value of concentration
IPFIX	Same as ITFIX, for pressure instead of temperature
P	Initial value of pressure
IWFIX	Same as ITFIX, for vapor fraction instead of temperature
W	Initial value of vapor fraction

## APPENDIX B: SAMPLE CYCLES

This appendix contains cycle diagram for five sample cycles provided with the present version of ABSIM in the subdirectory \DATA. The cycle diagrams have been printed as drawn on the computer screen, using the graphical interface. For the cycle B1, an input and an output file are also included. The following cycles are included:

- B1: SEC - Single-effect LiBr-water chiller
- B2: DECS1 - Double-effect LiBr-water chiller, series flow type 1
- B3: DECP - Double-effect LiBr-water chiller, parallel flow
- B4: DCCAP - Double-condenser-coupled, triple-effect LiBr-water chiller, parallel flow
- B5: GAX - Generator-absorber heat exchange, water-ammonia heat pump.

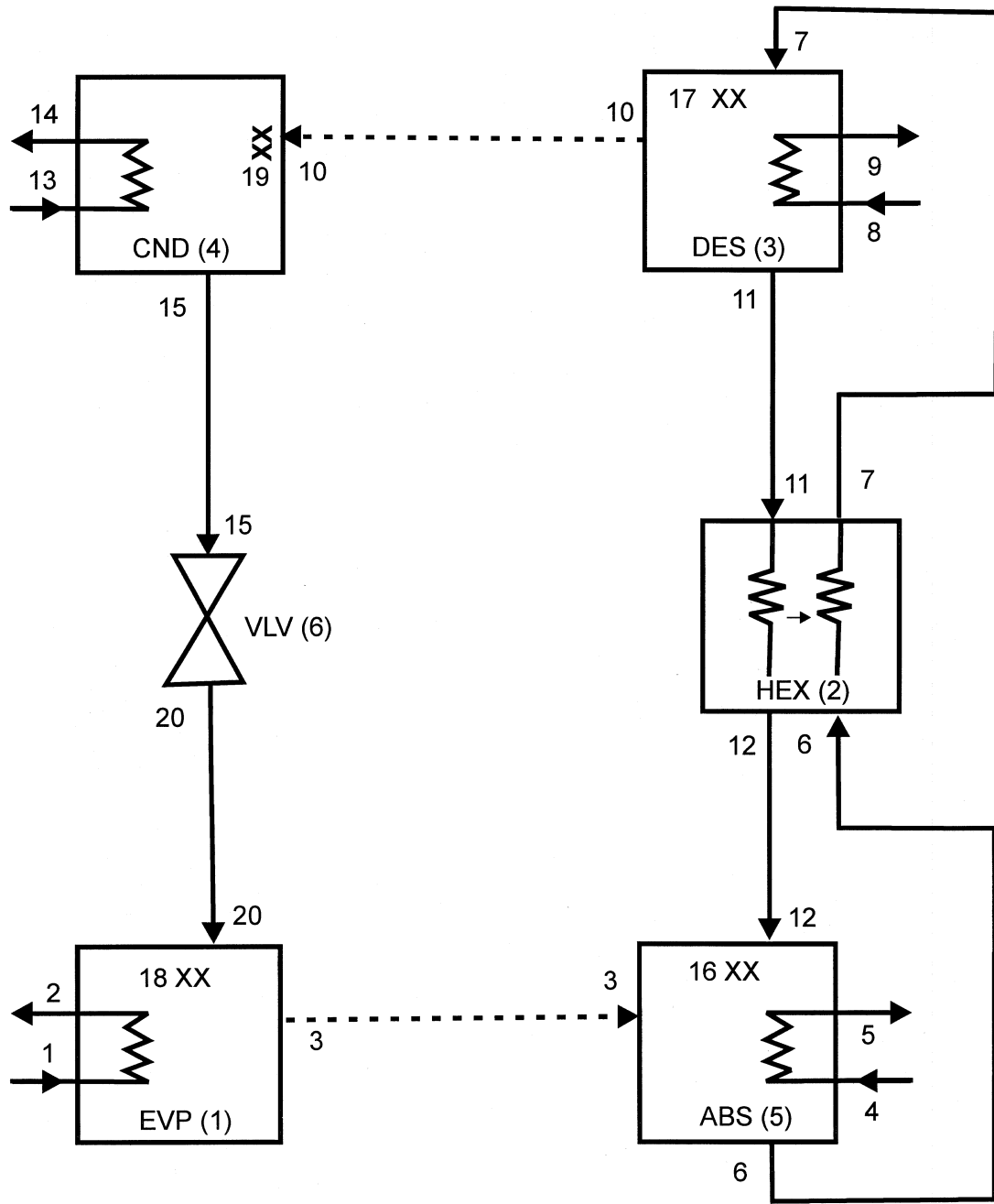


Fig. B.1: SEC – Single-effect LiBr-water chiller.



**Table B.1. Input file for single-effect LiBr-water chiller (SEC)**

---

```

LiBr-H2O 1-EFFECT CHILLER, CARRIER SAM-15
2.0000D+02 4.0000D+01 5.0000D+02 2.0000D+00
500 500 1 1.00D-05 1.00D-08 0
6 20
1 51 1 3.7700D+02 0 0.0000D+00 0.0000D+00 1
20 3 1 2 18 0 0
2 33 1 6.4000D+01 0 0.0000D+00 0.0000D+00 0
6 7 11 12 0 0 0
3 21 1 2.6800D+02 0 0.0000D+00 0.0000D+00 -1
7 10 8 9 11 17 0
4 41 1 5.6500D+02 0 0.0000D+00 0.0000D+00 0
10 15 13 14 19 0 0
5 11 1 1.9300D+02 0 0.0000D+00 0.0000D+00 0
12 3 4 5 6 16 0
6 61 0 0.0000D+00 0 0.0000D+00 0.0000D+00 0
15 20 0 0 0 0 0
1 3 1 1.0000D+02 0 3.0000D+02 0 0.0000D+00 0 0.0000D+00 0 0.0000D+00
2 3 0 4.5000D+01 0 3.0000D+02 0 0.0000D+00 0 0.0000D+00 0 0.0000D+00
3 3 3 1.0000D+02 2 2.5000D+00 0 0.0000D+00 3 1.0000D-01 0 1.0000D+00
4 3 0 8.5000D+01 0 4.8300D+02 0 0.0000D+00 0 0.0000D+00 0 0.0000D+00
5 3 1 1.0000D+02 0 4.8300D+02 0 0.0000D+00 0 0.0000D+00 0 0.0000D+00
6 1 1 1.0000D+02 0 6.0000D+01 3 5.7000D+01 3 1.0000D-01 0 0.0000D+00
7 1 1 1.0000D+02 0 6.0000D+01 3 5.7000D+01 2 1.0000D+00 0 0.0000D+00
8 3 0 1.8000D+02 0 4.1600D+02 0 0.0000D+00 0 0.0000D+00 0 0.0000D+00
9 3 1 1.0000D+02 0 4.1600D+02 0 0.0000D+00 0 0.0000D+00 0 0.0000D+00
10 3 2 1.0000D+02 2 2.5000D+00 0 0.0000D+00 2 1.0000D+00 0 1.0000D+00
11 1 2 1.0000D+02 3 5.7500D+01 2 5.9000D+01 2 1.0000D+00 0 0.0000D+00
12 1 1 1.0000D+02 3 5.7500D+01 2 5.9000D+01 3 1.0000D-01 0 0.0000D+00
13 3 0 8.5000D+01 0 3.9100D+02 0 0.0000D+00 0 0.0000D+00 0 0.0000D+00
14 3 1 1.0000D+02 0 3.9100D+02 0 0.0000D+00 0 0.0000D+00 0 0.0000D+00
15 3 4 1.0000D+02 2 2.5000D+00 0 0.0000D+00 2 1.0000D+00 0 0.0000D+00
16 1 1 1.0000D+02 1 6.0000D+01 1 5.9500D+01 3 1.0000D-01 0 0.0000D+00
17 1 1 1.0000D+02 1 6.0000D+01 1 5.6500D+01 2 1.0000D+00 0 0.0000D+00
18 3 3 1.0000D+02 2 2.5000D+00 0 0.0000D+00 3 1.0000D-01 2 1.0000D-01
19 3 4 1.0000D+02 2 2.5000D+00 0 0.0000D+00 2 1.0000D+00 0 1.0000D+00
20 3 3 1.0000D+02 2 2.5000D+00 0 0.0000D+00 3 1.0000D-01 2 1.0000D-01

```

---

**Table B.2: Output file for single-effect LiBr-water Chiller (SEC)**

LiBr-H2O 1-EFFECT CHILLER, CARRIER SAM-15

NORMALIZATION PARAMETERS (IN INPUT UNITS):

TMAX= .2000D+03 TMIN= .4000D+02 FMAX= .5000D+03 PMAX= .2000D+01

INPUT AND OUTPUT IN BRITISH UNITS

BRITISH UNITS ARE :

TEMPERATURES (T) IN DEG F  
 MASS FLOWRATES (F) IN LBS/MIN  
 CONCENTRATIONS (C) IN WEIGHT %  
 ENTHALPIES (H) IN BTU/LB  
 PRESSURES (P) IN PSIA  
 HEAT QUANTITIES (Q) IN BTU/MIN

TOLERANCES IN F, X : 1.0D-05 1.0D-08

NO. OF UNITS: 6

NO. OF STATE POINTS: 20

UNIT INPUT

1 51 1.3770D+03 0 .0000D+00 .0000D+00 1  
 20 3 1 2 18 0 0  
 2 33 1.6400D+02 0 .0000D+00 .0000D+00 0  
 6 7 11 12 0 0 0  
 3 21 1.2680D+03 0 .0000D+00 .0000D+00 -1  
 7 10 8 9 11 17 0  
 4 41 1.5650D+03 0 .0000D+00 .0000D+00 0  
 10 15 13 14 19 0 0  
 5 11 1.1930D+03 0 .0000D+00 .0000D+00 0  
 12 3 4 5 6 16 0  
 6 61 0.0000D+00 0 .0000D+00 .0000D+00 0  
 15 20 0 0 0 0 0

STATE POINT INPUT - STARTING VALUES

1 3 1 .1000D+03 0 .3000D+03 0 .0000D+00 0 .0000D+00 0 .0000D+00  
 2 3 0 .4500D+02 0 .3000D+03 0 .0000D+00 0 .0000D+00 0 .0000D+00  
 3 3 3 .1000D+03 2 .2500D+01 0 .0000D+00 3 .1000D+00 0 .1000D+01  
 4 3 0 .8500D+02 0 .4830D+03 0 .0000D+00 0 .0000D+00 0 .0000D+00  
 5 3 1 .1000D+03 0 .4830D+03 0 .0000D+00 0 .0000D+00 0 .0000D+00  
 6 1 1 .1000D+03 0 .6000D+02 3 .5700D+02 3 .1000D+00 0 .0000D+00  
 7 1 1 .1000D+03 0 .6000D+02 3 .5700D+02 2 .1000D+01 0 .0000D+00  
 8 3 0 .1800D+03 0 .4160D+03 0 .0000D+00 0 .0000D+00 0 .0000D+00  
 9 3 1 .1000D+03 0 .4160D+03 0 .0000D+00 0 .0000D+00 0 .0000D+00  
 10 3 2 .1000D+03 2 .2500D+01 0 .0000D+00 2 .1000D+01 0 .1000D+01  
 11 1 2 .1000D+03 3 .5750D+02 2 .5900D+02 2 .1000D+01 0 .0000D+00  
 12 1 1 .1000D+03 3 .5750D+02 2 .5900D+02 3 .1000D+00 0 .0000D+00  
 13 3 0 .8500D+02 0 .3910D+03 0 .0000D+00 0 .0000D+00 0 .0000D+00  
 14 3 1 .1000D+03 0 .3910D+03 0 .0000D+00 0 .0000D+00 0 .0000D+00  
 15 3 4 .1000D+03 2 .2500D+01 0 .0000D+00 2 .1000D+01 0 .0000D+00  
 16 1 1 .1000D+03 1 .6000D+02 1 .5950D+02 3 .1000D+00 0 .0000D+00  
 17 1 1 .1000D+03 1 .6000D+02 1 .5650D+02 2 .1000D+01 0 .0000D+00

18 3 3 .1000D+03 2 .2500D+01 0 .0000D+00 3 .1000D+00 2 .1000D+00  
 19 3 4 .1000D+03 2 .2500D+01 0 .0000D+00 2 .1000D+01 0 .1000D+01  
 20 3 3 .1000D+03 2 .2500D+01 0 .0000D+00 3 .1000D+00 2 .1000D+00

STARTING TEMPERATURES AFTER APPLYING CONSTRAINTS

1 100.0 1  
 2 45.0 0  
 3 45.0 3  
 4 85.0 0  
 5 100.0 1  
 6 100.0 1  
 7 100.0 1  
 8 180.0 0  
 9 100.0 1  
 10 100.0 2  
 11 100.0 2  
 12 100.0 1  
 13 85.0 0  
 14 100.0 1  
 15 100.0 4  
 16 100.0 1  
 17 100.0 1  
 18 45.0 3  
 19 100.0 4  
 20 45.0 3

NO. OF VARIABLES 23  
 NO. OF TEMPERATURES 12  
 NO. OF CONCENTRATIONS 4  
 NO. OF FLOWS 4  
 NO. OF PRESSURES 2  
 NO. OF VAPOR FRACTIONS 1

TOTAL NO. OF EQUATIONS 25  
 NO. OF NONLINEAR EQUATIONS 21  
 NO. OF LINEAR EQUATIONS 4  
 NO. OF REDUNDANT EQUATIONS 2

LAST ITERATION NO. IS 70  
 IER = 1  
 EUCLIDEAN NORM OF FUN IS LESS THAN FTOL

NO. VARIABLE RESIDUAL EQUATION

1	8.1222D-02	1.5850D-08	HEAT BALANCE,	UNIT NO. 1
2	1.1293D-02	6.4011D-06	HEAT TRANSFER,	UNIT NO. 1
3	3.2289D-01	-1.3513D-07	HEAT BALANCE,	UNIT NO. 2
4	3.7775D-01	1.4874D-08	HEAT TRANSFER,	UNIT NO. 2
5	6.6408D-01	-1.5189D-06	T5=T5E(P2,C5)+DEVL	DESORBER NO. 3
6	8.2488D-01	-1.4335D-06	T6=T6E(P2,C6)	DESORBER NO. 3
7	8.0324D-01	-1.5565D-08	ADIABATIC EQB.	DESORBER NO. 3
8	4.9890D-01	-3.9563D-10	FLOW AT INTRNL PT.	DESORBER NO. 3
9	3.2186D-01	4.3298D-08	HEAT BALANCE,	UNIT NO. 3
10	3.3434D-01	2.3074D-06	HEAT TRANSFER,	UNIT NO. 3
11	4.3479D-01	-1.3813D-06	T2=T2E(P1,C2)+DEVL	CONDNSER NO. 4

12	7.4031D-01	-5.6124D-09	HEAT BALANCE,	UNIT NO. 4
13	5.6112D-01	-1.7635D-07	HEAT TRANSFER,	UNIT NO. 4
14	5.8411D-01	5.7646D-07	T5=T5E(P2,C5)+DEVL	ABSORBER NO. 5
15	5.8657D-01	4.7388D-07	T6=T6E(P2,C6)	ABSORBER NO. 5
16	5.5818D-01	-1.9963D-08	ADIABATIC EQB.	ABSORBER NO. 5
17	4.7230D-03	-8.0848D-09	FLOW AT INTRNL PT.	ABSORBER NO. 5
18	1.1528D-01	8.8867D-08	HEAT BALANCE,	UNIT NO. 5
19	1.1479D-01	9.5955D-08	HEAT TRANSFER,	UNIT NO. 5
20	1.2063D-01	4.5485D-07	T2=T2E(P2,C2)	VALVE NO. 6
21	6.5273D-02	-2.1345D-07	H(IN)=H(OUT)	VALVE NO. 6
22	3.8950D-01	8.4981D-15	TOTAL MASS BALANCE,	DESORBER NO. 3
23	4.8318D-02	-1.0292D-08	ABS. MASS BALANCE,	DESORBER NO. 3
24	-1.1353D-01	-8.4981D-15	TOTAL MASS BALANCE,	ABSORBER NO. 5
25	4.3552D-02	1.0292D-08	ABS. MASS BALANCE,	ABSORBER NO. 5

STATE TEMPER. ENTHALPY FLOW RATE CONCENTR. PRESSURE VAPOR FR.  
POINT

1	5.2996D+01	2.1063D+01	3.0000D+02	0.0000D+00	0.0000D+00	0.0000D+00
2	4.5000D+01	1.3050D+01	3.0000D+02	0.0000D+00	0.0000D+00	0.0000D+00
3	4.1807D+01	1.0794D+03	2.3615D+00	0.0000D+00	1.3055D-01	1.0000D+00
4	8.5000D+01	5.3045D+01	4.8300D+02	0.0000D+00	0.0000D+00	0.0000D+00
5	9.1662D+01	5.9695D+01	4.8300D+02	0.0000D+00	0.0000D+00	0.0000D+00
6	1.0044D+02	4.0270D+01	6.0000D+01	5.6112D+01	1.3055D-01	0.0000D+00
7	1.4625D+02	6.2447D+01	6.0000D+01	5.6112D+01	7.7901D-01	0.0000D+00
8	1.8000D+02	1.4801D+02	4.1600D+02	0.0000D+00	0.0000D+00	0.0000D+00
9	1.7198D+02	1.3997D+02	4.1600D+02	0.0000D+00	0.0000D+00	0.0000D+00
10	1.6852D+02	1.1354D+03	2.3615D+00	0.0000D+00	7.7901D-01	1.0000D+00
11	1.6852D+02	7.6506D+01	5.7639D+01	5.8411D+01	7.7901D-01	0.0000D+00
12	1.1982D+02	5.3420D+01	5.7639D+01	5.8411D+01	1.3055D-01	0.0000D+00
13	8.5000D+01	5.3045D+01	3.9100D+02	0.0000D+00	0.0000D+00	0.0000D+00
14	9.1497D+01	5.9531D+01	3.9100D+02	0.0000D+00	0.0000D+00	0.0000D+00
15	9.3495D+01	6.1525D+01	2.3615D+00	0.0000D+00	7.7901D-01	0.0000D+00
16	1.0957D+02	4.9100D+01	5.7397D+01	5.8657D+01	1.3055D-01	0.0000D+00
17	1.5845D+02	6.8066D+01	6.0316D+01	5.5818D+01	7.7901D-01	0.0000D+00
18	4.1807D+01	6.1525D+01	2.3615D+00	0.0000D+00	1.3055D-01	4.8318D-02
19	9.3495D+01	1.1020D+03	2.3615D+00	0.0000D+00	7.7901D-01	1.0000D+00
20	4.1807D+01	6.1525D+01	2.3615D+00	0.0000D+00	1.3055D-01	4.8318D-02

NO.	TYPE	UA	NTU	EFF	CAT	LMTD
1	EVAPORATOR	3.7696D+02	1.2539D+00	7.1461D-01	3.1931D+00	6.3765D+00
2	HEAT EXGER	6.4000D+01	2.3421D+00	7.1528D-01	1.9384D+01	2.0791D+01
3	DESORBER	2.6799D+02	8.0688D-01	4.6722D-01	1.1481D+01	1.2478D+01
4	CONDENSER	5.6500D+02	1.4476D+00	7.6487D-01	1.9973D+00	4.4883D+00
5	ABSORBER	1.9300D+02	5.4837D-01	3.7148D-01	1.5440D+01	1.6642D+01
6	VALVE	0.0000D+00	0.0000D+00	0.0000D+00	0.0000D+00	0.0000D+00

NO.	TYPE	HEAT TRANSFER	DEVL	DEVG	IPINCH
1	EVAPORATOR	2.4037D+03	0.000D+00	0.000D+00	-1
2	HEAT EXGER	1.3306D+03	0.000D+00	0.000D+00	-1
3	DESORBER	3.3441D+03	0.000D+00	0.000D+00	1
4	CONDENSER	2.5359D+03	0.000D+00	0.000D+00	1
5	ABSORBER	3.2119D+03	0.000D+00	0.000D+00	-1
6	VALVE	0.0000D+00	0.000D+00	0.000D+00	0

COP = .7188      CAPACITY = 2.4037D+03 BTU/MIN

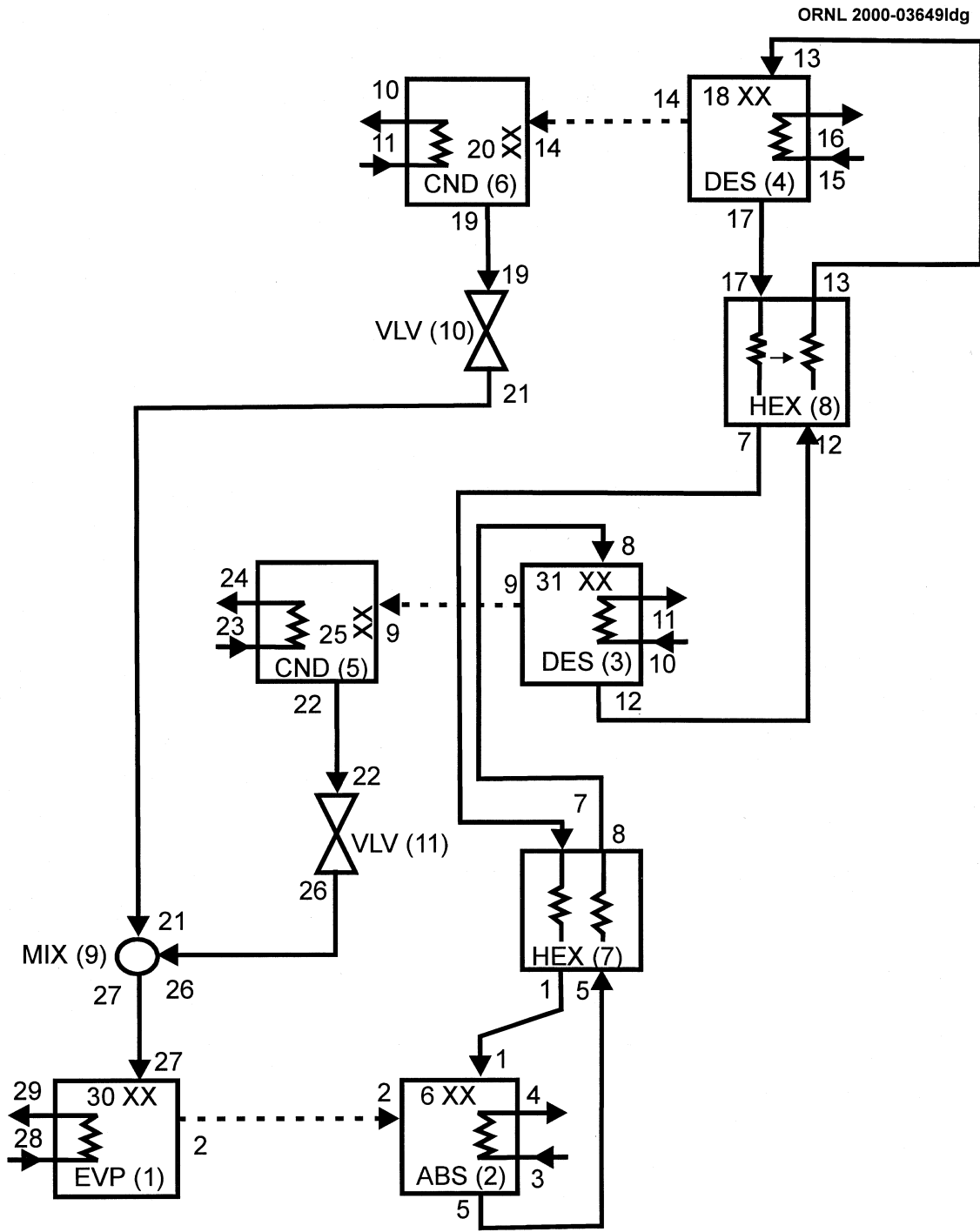


Fig. B.2: DECS1—Double-effect LiBr-water chiller, series flow type 1.

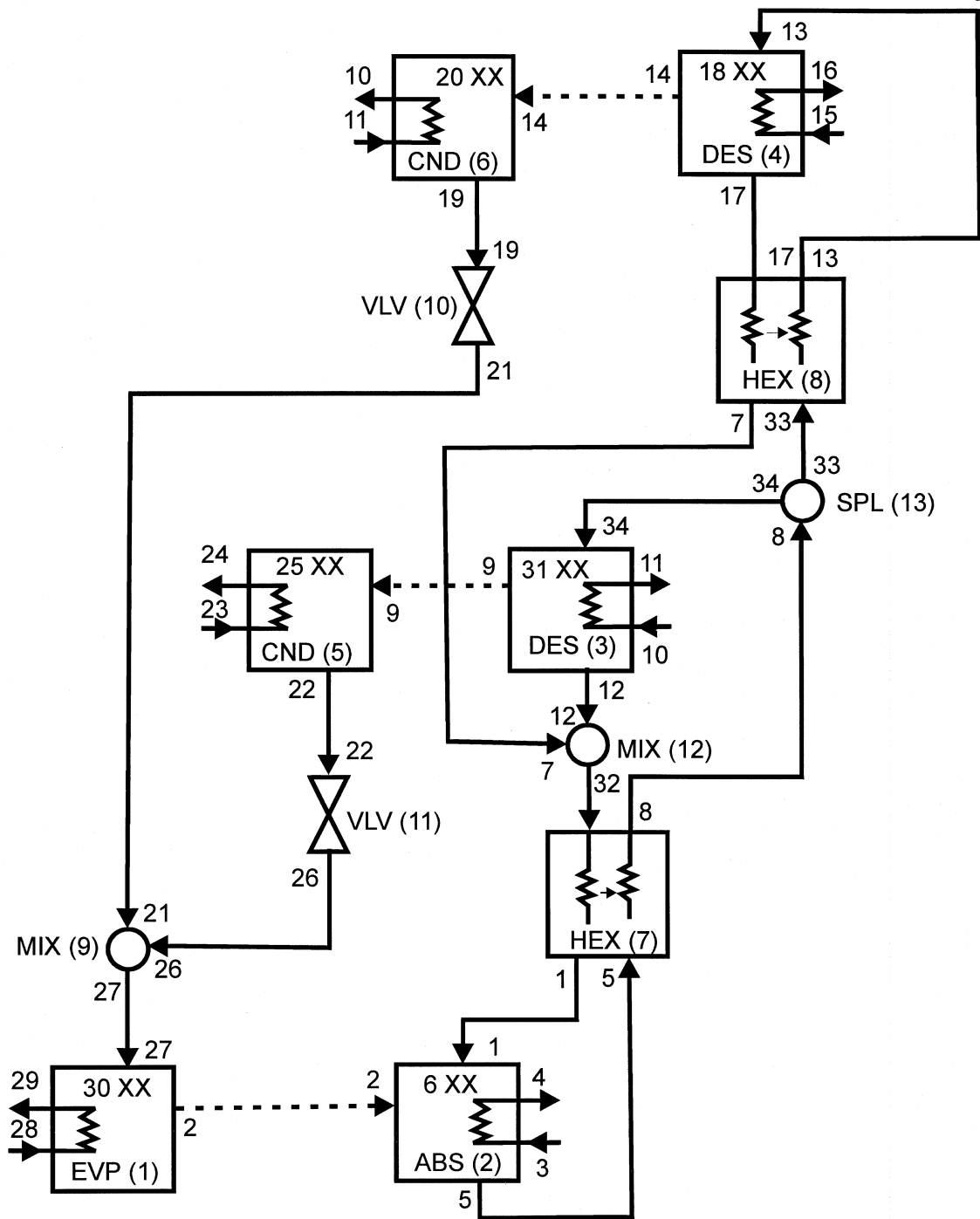
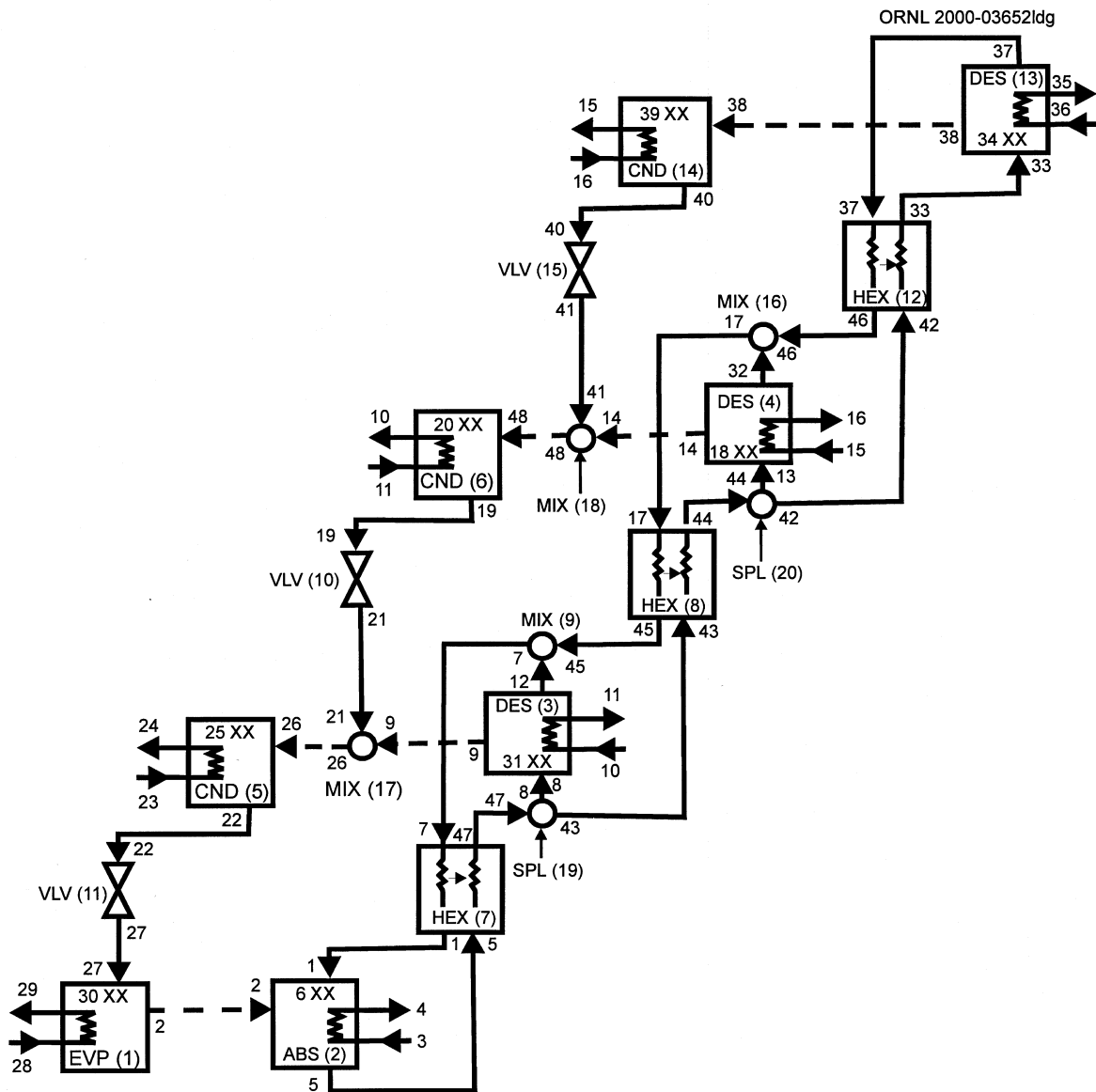


Fig. B.3: DECP—Double-effect LiBr-water chiller, parallel flow



**Fig. B.4: DCCAP—Double-condenser-coupled, triple-effect  
LiBr- water chiller, parallel flow.**

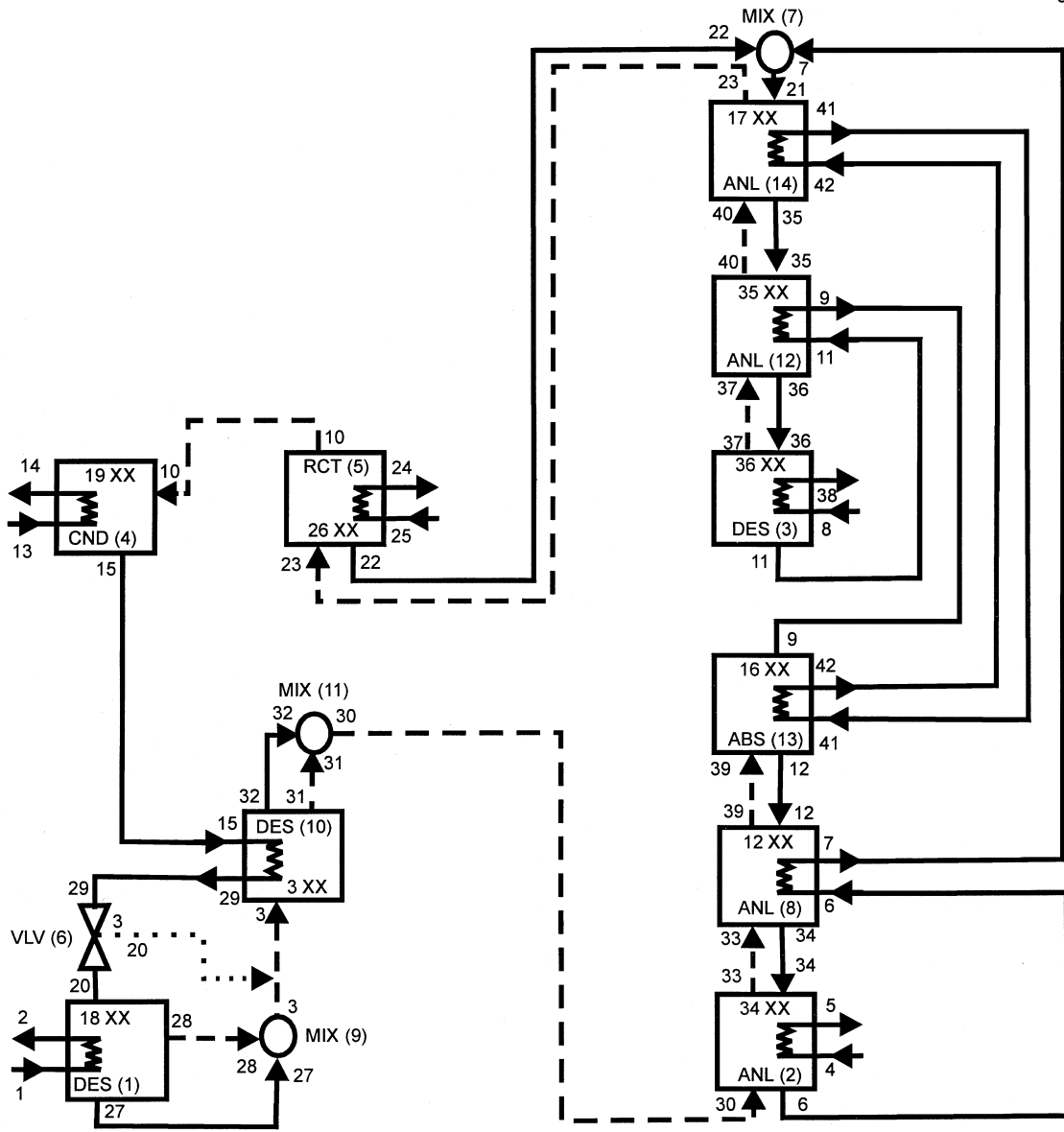


Fig. B.5: GAX—Generator-absorber heat exchange, water-ammonia heat pump.



**INTERNAL DISTRIBUTION**

- |                    |                              |
|--------------------|------------------------------|
| 1. J. E. Christian | 8. W. A. Miller              |
| 2. G. E. Courville | 9. C. I. Moser               |
| 3. T. R. Curlee    | 10. A. C. Schaffhauser       |
| 4. R. C. DeVault   | 11. R. B. Shelton            |
| 5. P. D. Fairchild | 12–17. A. Zaltash            |
| 6. S. K. Fischer   | 18. RC                       |
| 7. P. W. Garland   | 19. Central Research Library |

**EXTERNAL DISTRIBUTION**

20. Lilia A. Abron, PEER Consultants, P.C., 1460 Gulf Blvd., Apt. 1103, Clearwater, FL 33767
21. G. Anand, Energy Concepts Co., 627 Ridgely Avenue, Annapolis, MD 21401
22. Douglas Bauer, Commission on Engineering and Technical Systems, National Research Council, Harris 280, 2001 Wisconsin Ave. NW, Washington, D.C. 20007
23. W. Biermann, 110 Foxcroft Drive, Fayetteville, NY 13066
24. H. Perez-Blanco, Department of Mechanical Engineering, The Pennsylvania State University, 338B Reber Building, University Park, PA 16802-1414
25. A. T. Conlisk, Department of Mechanical Engineering, The Ohio State University, 206 West 18th Avenue, Columbus, OH 43210-1107
26. Susan L. Cutter, Hazards Research Laboratory, Department of Geography, University of South Carolina, Columbia, SC 20208
27. T. E. Drabek, Department of Sociology, University of Denver, Denver, CO 80208-0209
28. D. C. Erickson, Energy Concepts Co., 627 Ridgely Avenue, Annapolis, MD 21401
29. R. J. Fiskum, U.S. Department of Energy, EE-422, 1000 Independence Ave., SW, Washington, D.C. 20585-0121
- 30–36. G. Grossman, Mechanical Engineering, Technion—Israel Institute of Technology, Haifa 32000, Israel
37. W. T. Hanna, Battelle-Columbus, 505 King Avenue, Columbus, OH 43201-2693
38. S. G. Hildebrand, Oak Ridge National Laboratory, P.O. Box 2008, Oak Ridge, TN 37831-6037

39. J. A. Kohler, York International Corporation, P.O. Box 1592-191A, York, PA 17405-1592
40. B. A. Phillips, Phillips Engineering Company, 721 Pleasant Street, St. Joseph, MI 49085
41. R. Radermacher, University of Maryland, Mechanical Engineering Department, College Park, MD 20742
42. P. R. Rittelmann, Burt Hill Kosar Rittelmann Associates, 400 Morgan Center, Butler, PA 16001-5977
43. U. Rockenfeller, Rocky Research, 1598 Foothill Drive, Boulder City, NV 89005
44. W. A. Ryan, Gas Research Institute, 8600 West Bryn Mawr Avenue, Chicago, IL 60631
45. C. T. Sgamboti, United Technologies Research Center, 411 Silver Lane, MS 129-79, East Hartford, CT 06108
46. K. Schultz, The Trane Company, 3600 Pammel Creek Road, La Crosse, WI 54601-7599
47. S. F. Tierney, The Economic Research Group, 1 Mifflin Place, Cambridge, MA 02138
48. C. M. Walton, University of Texas at Austin, Austin, TX 78712-1076
- 49-50. OSTI, U.S. Department of Energy, P.O. Box 62, Oak Ridge, TN 37831