NOTICE: This document contains information of a preliminary nature and is not intended for release. It is subject to revision or correction and therefore does not represent a final report.

ORNL/TM-XXXX/XX

Sorption Simulation Software v1.0 User's Guide and Reference

Prepared by
Zhiyao Yang
Dr. Ming Qu

March 2015

Draft. Not for public release.

OAK RIDGE NATIONAL LABORATORY
MANAGED BY UT-BATTELLE FOR THE US DEPARTMENT OF ENERGY

ORNL/TM-XXXX/XXX

Sorption Simulation Software
User's Guide and Reference

Date Published: March 2015

Prepared by Zhiyao Yang
and Dr. Ming Qu
for the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, TN 37831-6283
managed by
UT-BATTELLE, LLC
for the
US DEPARTMENT OF ENERGY
under contract DE-AC05-00OR22725

Table of Contents

1. OVERVIEW

1.1. WELCOME TO SORPSIM

SorpSim (Sorption system Simulation program) is a user-friendly computer software developed based on ABSIM, a program initially developed by the Oak Ridge National Laboratory between 1980 and 1990. ABSIM was originally developed for the performance prediction of steady-state closed-cycle absorption systems. It provides a platform for investigating various cycle configurations of absorption systems by using different working fluids. SorpSim inherits the expertise of modular simulation of absorption systems from ABSIM and further expands the capabilities by adding modules for liquid desiccant systems. SorpSim can be used on various computer operating systems such as Windows and Mac OS X. It has a brand new user friendly Graphical User Interface (GUI) for users to investigate the system cycles via various new functions like system level data review/edit panel, parametric tables, debugging information, and charts. The calculation engine of SorpSim is based on ABSIM Windows Version 5.0's.

The GUI of SorpSim enables users to build the cycle diagram with simple drag-drop and linking actions, and define the case data interactively. SorpSim constructs a case using a special data structure to synchronize all the data and settings provided by users. The program calculates the unknown operating parameters at each state point, as well as the heat duty at each component. The system capacity and performance parameters can be determined. The GUI also provides powerful analysis functionalities for users to view the cycle on property charts and carry out parametric studies using tables and plots.

This user manual is organized into five sections with two appendix sections. The remainder of this section describes the background for the development of SorpSim in comparison with other available simulation tools for sorption systems. Section 2 contains information of software installation and basic operations for the first-time user. Section 3 describes the structure of the software, including the relationship of the integrated calculation engine and the SorpSim GUI, an introduction to the GUI development, and the data structure and database, which SorpSim utilizes to manage and store case data. Section 4 describes the subroutines containing the governing equations for the basic 12 types of the components inherited from ABSIM, plus three expanded liquid desiccant components. Section 5 introduces the functions and features of the GUI, and instructs users on how to use this interactive interface. Appendix A introduces the

tools and libraries used for SorpSim development. Appendix B contains brief information about several example cycles available in the SorpSim.

1.2.BACKGROUND

Absorption systems, particularly absorption heat pumps, have drawn renewed interest during the past years due to their advantage of relative noise-and-vibration-free operation and the possibility to use low-grade heat source like solar and waste heat to energize the system [1]. Such increase of research and design on absorption system has created a need for reliable and effective system simulations. Several computer models had been developed for research and design optimization of some particular absorption systems. The software tools most often referenced in the literature for absorption modeling include Engineering Equation Solver (EES), ASPEN Plus and ABSIM.

EES is an equation-oriented code with built-in properties routines that allow users to compute thermo-physical properties of the two most common working fluid pairs suitable for absorption systems and heat pumps [2], and absorption simulations have been conducted using EES [3]. However, in EES all governing equations of the system must be listed and initial guess value be properly defined by users, which makes modeling of cycles dependent on detailed user knowledge and susceptible to user errors. Moreover, due to non-linear equations involved in absorption cycle models, the results have to be checked for physical feasibility. Thus users would have to do a fair amount of programming and adjustment of variables to achieve convergence and the correct solution.

ASPEN Plus is a steady-state modeling software package that is widely used by the chemical industry for process simulation. It provides the flexibility for various cycle configurations and different working fluids, and is used in some cases for absorption simulation during the past decade [4-6]. ASPEN Plus has an excellent GUI that helps users to configure and define the system cycle and diagnose errors. The models of simple components and extensive libraries of equations of state (EOS) and working fluids in ASPEN Plus support robust operation for a wide range of cycle configurations. However, such features also introduce complexity to simulation of a particular type of system. For absorption system models developed in ASPEN Plus, users would have to specify different EOS for particular scenarios. Moreover, many essential components in absorption system such as desorber are not provided by the ASPEN Plus, and

users would have to add and define several simple components to form a macro that can function as one component in absorption system. Thus simulating with this software requires considerable time and training investment from users before being able to employ the extensive libraries of modules and working substances, as well as to deal with the subjectivity of breaking a process into simpler “blocks” before simulation.

ABSIM was developed during the 1980s and 1990s with the objective of creating a simulation tool for evaluating absorption systems in varying cycle configurations and with different working fluid [7]. The original version of ABSIM achieved flexibility without the need for a user-defined iterative solution scheme through a modular approach[8, 9]. After its latest update in 2000, it was capable of accurate simulations of advanced and complex cycles, and was equipped with a graphical user interface (GUI) for input preparation and output analysis [10]. Being a code specially designed for absorption system simulation, ABSIM was a powerful tool for convenient model construction while yielding validated and accurate results. Thus it was widely used for simulations of absorption systems with various cycle configurations and working fluid pairs during the 1990s.

However, this version of ABSIM (ABSIMW 5.0) has been less popular for investigation of absorption systems during the past decade due to some limitations: it is not compatible with most current mainstream operating systems (requiring Windows XP or earlier Windows OS); a modern update could make important improvements to the user-friendliness of system and parameter setting and debugging tools to help achieve convergence.

The SorpSim has been developed on this background to provide a user-friendly computer simulation tool with powerful functionalities to build, simulate, and carry out further analysis for sorption systems in different cycle configurations and with different working fluids. The software has been developed using Qt/C++ framework to provide great graphical features as well as unmatched support for multiple operating systems. The SorpSim is specialized in simulating absorption system thanks to the powerful calculation engine and well-established component and fluid property libraries that it inherited from the ABSIM. The SorpSim is also well capable of dealing with liquid desiccant systems with the new desiccant component models implemented to expand the component library. Users don’t have to do any programming as the equation and variable vector are all automatically constructed along with the graphical

operations on the GUI. The extensive use of xml file as database of the calculation cases in the SorpSim allows further analysis been carried out after simulation of a single case is done.

2. GETTING STARTED

2.1. SORPSIM INSTALLATION

SorpSim is distributed through an installer for Windows platform from Windows XP to the latest Windows 10, or an App package for Mac machines running OS X. After installation, the software folder/package contains the program executable, all the dynamic linked libraries associated to the software, the files containing template/example cases, and other supporting information.

To install the SorpSim on Windows machines, users only need to double-click the installer and follow the steps. During the installation, users would be asked to specify the directory where all the files of the software will be located. The installed software can be found in a folder named "SorpSim package" under the designated directory.

To install the SorpSim on the OS X platforms, users would only need to copy the distributed software package to a local directory, as all the associated files are included in the package.

2.2.RUNNING SORPSIM

Upon launching the software, the main window of the SorpSim will show along with a starting dialog. The starting dialog guides users to creating a new case project or resuming to a previous case project.

The main menu of the SorpSim main window includes the following entries:

- File —to manage the SorpSim case files, import an ABSIM case file, and print/export the cycle diagram
- Edit—to manipulate the components on the cycle diagram, control the operating panel view, and access the parameters of system/component/state-point level
- Construct—to create a new component/link/custom-text-item, define its parameters and contents, and add it onto the cycle diagram
- Calculate—to review and edit additional correlations, initiate simulation of the current case, view convergence information, and display/export the simulation results in various forms

- Tables—to create a new parametric table based on the current system cycle, and to access the existing tables
- Plots—to create a new parametric plot based on an existing parametric table, to create a new property chart to help analyze the cycle, and to access the existing plots
- Help—to access the help information about software functions and component models, access the information about the software, and to load example cases provided with the software

Below the menu bar there is a tool bar for quick access to some of the most frequently used functions.

For more detailed instructions on each item of the menu, users are recommended to refer to Section 5 of this manual, where the SorpSim GUI is described in details. However, the program is quite user-friendly, and most of the entries are self-explanatory. Users are encouraged to explore and attempt to perform some basic operations using the example cases provided with the software. Users can start by going to the Help menu, then choose one of the example files from the menu, and define the unit system for the loaded case. Run the simulation by going to Calculate→Run and initiate the calculation using default settings. Display the results on the cycle diagram once the calculation is finished. Then try to save the case into your own case file by going to File→Save and specify the directory and file name. The parameters of components and state points can be accessed by double-clicking the item of interest. Be there any confusion about menu functions, go to Help→Content and view the help documentation of each function on the menu.

3. STRUCTURE OF THE SOFTWARE

The SorpSim Framework is shown in Figure 3-1. The core of the software is consisted of the SorpSim GUI and the Calculation Engine. The GUI manages case data stored in XML file databases, and it also provides an interactive environment for users to configure the case. Once the simulation task is initiated by users in the GUI, case data is packaged in special data structure and sent to the Calculation Engine as input. The Calculation Engine solves the case by interpreting the input into variable vectors, and calling subroutines in the component model library and fluid property library. The results are generated and sent back to the GUI, where they are displayed to users and stored in the case data.

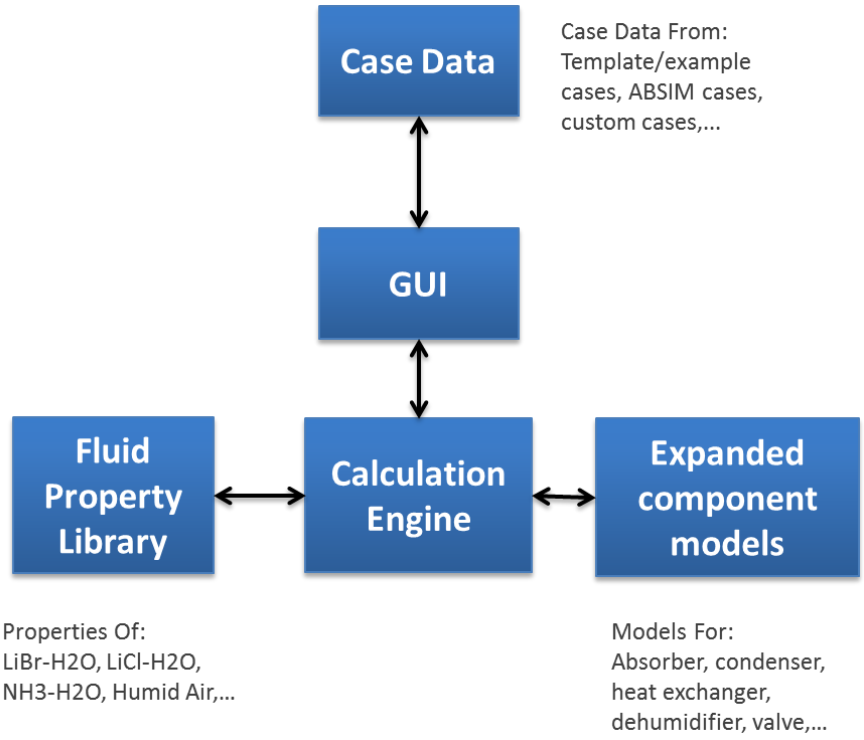

Figure 3-1: SorpSim Framework

As shown in the framework, the SorpSim program can be divided into two core sections: the GUI which focuses on managing the case data and interacting with users, and the calculation engine that simulates each case sent by the GUI, and presents the results back to the GUI. The calculation engine along with the property library and the absorption component model library is inherited from the ABSIM and integrated as a module into the GUI program.

In this section, the integration of the calculation engine into the GUI program is first described, followed by the data structure and database that are implemented in the GUI program to manage and store case data. The tools that have been used for the SorpSim development are introduced in Appendix A.

3.1. INTEGRATION OF THE CALCULATION ENGINE

The calculation engine as part of the core of the SorpSim software was inherited from the ABSIMW 5.0 program. After converting the source code of the ABSIM's calculation engine into the same language as the GUI development, the calculation engine becomes a large C++ class with all the functions and subroutines as its member methods. The original program flow of the calculation is preserved during conversion, which means simulation can still be initiated with case data prepared in proper format, and the results will be prepared to the output.

Originally in ABSIMW 5.0, the input case data and the output result data are communicated between the GUI and the calculation engine via text-based files [7]. This is not safe or efficient from the modern software's view, thus the first thing for the integration is to implement in-program data communication mechanism.

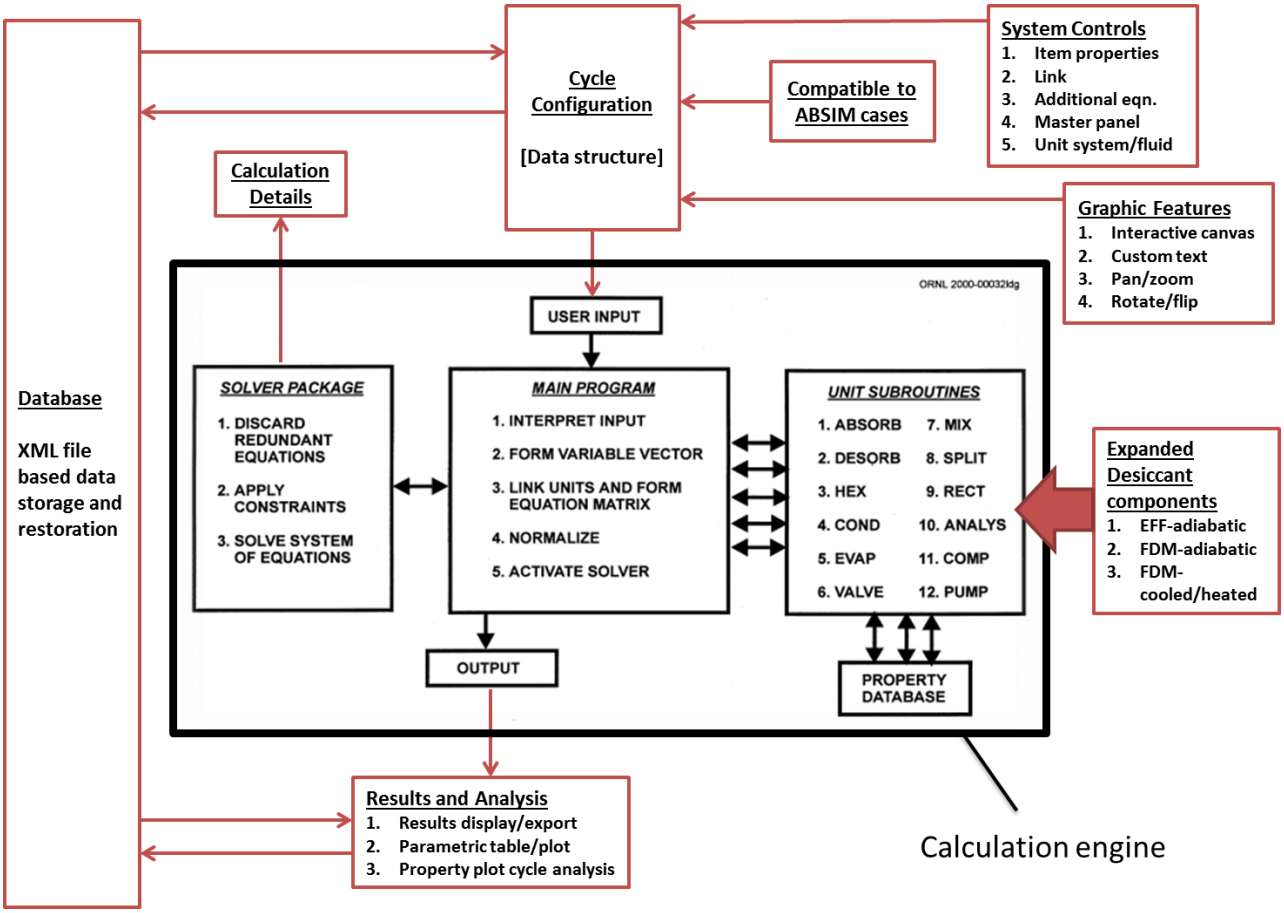

Figure 3-2. Data communication in GUI and the calculation engine

Figure 3-2 illustrates the data communication between different sections of the GUI and the calculation engine. The blocks inside the black box are the calculation engine. Although written in a different language, the structure and program flow of the calculation engine is the same as the original engine of ABSIMW 5.0. Descriptions of the calculation engine structure can be found in the user manual of ABSIMW 5.0 [7]. The information about property database and the solver package can also be found with details in the ABSIMW 5.0 user manual, as the SorpSim directly use the code concerning these two parts, it is recommended that questions related to these parts are addressed referring to the ABSIM's manual. For working fluid used in liquid desiccant systems, several additional subroutines are implemented as well. The additional property subroutines and their reference can be found in Table 1.

Table 1 additional fluid property subroutines for liquid desiccant systems

Desiccant	Property	Subroutine	Reference
LiCl-water	Specific heat	CPFTX9	Conde [11]
	Equilibrium humidity ratio	WFTX9	Conde [11]
	Enthalpy	HFTX9	Chaudhari [12]
LiBr-water	Specific heat	CPFTX1	Iyoki and Uemura [13]
	Equilibrium humidity ratio	WFTX1	ASHRAE handbook [14]

The items outside the black box are different parts of the GUI program, except for the desiccant component modules, which are implemented within the component model library inside the calculation engine (more details in Section 4).

As shown in Figure 3-2, the data communication between GUI and the calculation engine includes:

1> The GUI interprets the graphical system configuration on the cycle diagram and combines it with all the parameters and settings to generate a complex data object containing the entire set of data for current case. Once users initiate calculation, this data object will be tunneled to the calculation engine as the user input.

2> The calculation engine then gathers these data and constructs variable vectors and equation matrixes as the preparation to activate the solver. Once simulation is finished, the results containing all status parameters of state points, component, and the entire system are gathered and stored in a different data object than the input, before the GUI takes the calculation outcomes via the data object and uses them as displayed results, or as bases for parametric and cycle analysis.

3> During calculation, the calculation engine generates messages and keeps track of the guess values of each iteration as well as the function residuals. This information is gathered and communicated to the GUI once the calculation engine stopped work, so that some useful information can be provided about the failed calculation if convergence was not achieved.

By using these 3 data communication mechanisms, the GUI is able to fully control the operation of the calculation engine to schedule simulation of a single case or multiple cases during parametric study, to dispatch and gather data seamlessly, and to monitor the process of the calculation. In this way, the SorpSim is able to inherit all the calculation capability as well as the component and fluid flexibility of the ABSIMW 5.0 program, and can be used to simulate absorption cycles with much convenience and reliability. With the liquid desiccant expansion to the component model library, the SorpSim's capacity of calculation is even further stretched to desiccant and hybrid systems.

3.2.DATA STRUCTURE AND DATABASE IN THE SORPSIM GUI

The SorpSim GUI offers a platform where users can define system cycles of any configuration modularly. This is supported by the data structure and database behind the interface. The data structure is designed to achieve modular composition of the system cycle and easy access to data. A database is also implemented to store all the information needed to retrieve a system cycle and parametric table.

In Figure 3-2, the data communication between the Cycle Configuration and other different parts of the GUI is realized based on the data structure that collects the case data that users insert at various time during system configuration, and at different dialogs and locations within the GUI.

The data structure is designed based on the behavior of components and state points in system cycle to achieve corresponding modification and easy access of data. For a system cycle constructed with modular components, the number and types of components are not pre-set. Thus linked-list is selected as the data structure for components for the simplicity it provides to append and modify data members in the middle of the structure. State points are added to and removed from the system with the components they are affiliated to. Thus state point data is stored in arrays that are child members of components' linked-list data structure. Figure 3-3 describes schematically the structure of the linked-list-array mixed data structure. When a new component is added into the system, it is appended at the end of the linked-list. Index numbers for this component and the state points inside it is automatically assigned. When access to a particular parameter is needed, the program will search through the linked list to the desired component, and retrieve data either directly from the component or from inside state points.

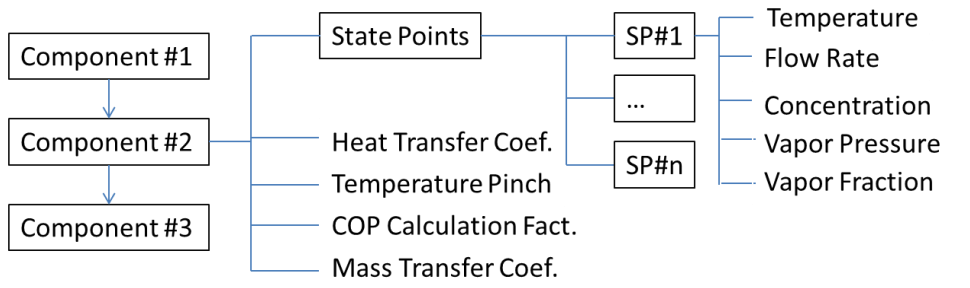

Figure 3-3: data structure for components and state points

The data structure containing system cycle configuration information can be accessed and modified by the system control elements in the GUI, such as item properties dialogs to set up component and state point parameters, links that merge two state points into one, additional equations that provide additional equal-relationship between unknown state point parameters. Any changes on the interface about the cycle data are synchronized with the backstage data structure. Actions such as adding a new component or breaking the link between two state points will trigger a set of corresponding changes to the values in the data structure, or the dimension of the data structure itself.

Unlike the data structure, which acts like a workshop where all the changes are instantly enforced and content updated, the database of the SorpSim is used to store the case data and other information so that they can be organized into an entity for keeping, and later conveniently retrieved as a whole. The database the SorpSim utilizes is the eXtensible Markup Language, or XML files. By applying a set of rules for encoding documents in a layer-based format, and applying the method to parse through the file, a large amount of data with tree-shaped relationship can be easily organized, stored, and accessed.

Since the data structure used by the SorpSim is a tree-shaped structure, it is convenient to pack up the content of the data structure into layers in the XML file, as well as to extract all the data in a properly formatted XML file and fill it into the data structure. The function of saving and loading system cases are implemented by bridging the data in the current system with the data stored in an XML file. The database is structured in three layers for system, components and state points so that regardless of the configuration of the system cycle, the program will be able to re-create a case exactly like the saved one with all pre-set parameters.

Another usage of the database is in parametric analysis functions. Parametric analysis in the new ABSIM also takes advantage of the XML database. For each table, the data is stored in three layers of runs, input/output and value. Information structured in this format makes it very convenient to locate and access some specific values of interest given the "address" of the value. Such method with data parsing in the XML file is used to retrieve existing parametric table, update table form and value changes, and provide data to draw the parametric plot.

Based on the integrated simulation engine and these specially designed data structure and database, the SorpSim is able to manage large amount of data from user-defined system cycles, and take advantage of the excellent capability of calculation and flexibility of cycle configuration of the ABSIM program.

4. COMPONENT MODULES

The component model library is included in the ABSIMW 5.0's source code package. The component subroutines in the library contain all the governing equations needed to describe the conservation relations, as well as the heat and mass transfer process. In this section, a complete list of component models, including the absorption components that were provided in the ABSIM package, and the desiccant components that are added to SorpSim. The absorption component models are originally described in the ABSIMW 5.0 manual [7].

4.1.ABSORPTION COMPONENTS

4.1.1. Absorber

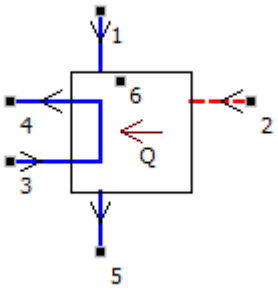

Fig. 4-1 Absorber

The absorber is referred to in the code as component type 1 and is described schematically in the Figure 4-1.

A stream of strong liquid absorbent (state 1) enters the component 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 weaker absorbent exits at state 5. The entering, strong absorbent may be subcooled or superheated; it is therefore allowed to reach equilibrium at point 6 (point inside the component) either by adiabatic absorption or desorption 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,

$$F_1 + F_2 - F_5 = 0$$

- Absorbent/absorbate mass balance,

$$F_1 C_1 + F_2 C_2 - F_5 C_5 = 0$$

- Energy balance,

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

- Heat transfer [one of the following equations]

$$Q_{absorber} = F_3 (h_4 - h_3)$$

$$LMTD = \frac{(T_6 - T_4) - (T_5 - T_3)}{ln[\frac{T_6 - T_4}{T_5 - T_3}]}$$

$CAT = (T_5 - T_3)$ *For IPINCH = -1 (temperature pinch at cold side)*

$CAT = (T_6 - T_4)$ *For IPINCH = 1 (temperature pinch at hot side)*

$EFF = \frac{T_6 - T_5}{T_6 - T_3}$ *For IPINCH = -1 (temperature pinch at cold side)*

$EFF = \frac{T_4 - T_3}{T_6 - T_3}$ *For IPINCH = 1 (temperature pinch at hot side)*

- Mass transfer

$$T_5 = T_{5E}(P_5, C_5) + DEVL = 0$$

- Adiabatic equilibrium at point 6

$$If\ T_6 = T_1, then\ C_6 = C_1\ and\ F_6 = F_1$$

$$\frac{h_1 - h_2}{h_1 - h_6} = \frac{C_1 - C_2}{C_1 - C_6}$$

$$T_6 = T_{6E}(P_6, C_6) = 0$$

$$\frac{F_6}{F_1} = \frac{C_1 - C_2}{C_6 - C_2}$$

The constraints are:

$$T_6 > T_4$$
$$T_5 > T_3$$
$$T_4 \geq T_3$$

4.1.2. Desorber

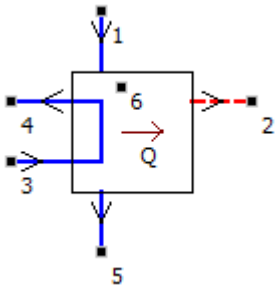

Fig. 4-2 Desorber

The desorber is referred to in the code as component type 2 and is described schematically in figure 4-2.

A stream of weak liquid absorbent (state 1) enters the component 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 inside the component) either by adiabatic absorption or desorption 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,

$$F_2 + F_5 - F_1 = 0$$

- Absorbent/absorbate mass balance,

$$F_5 C_5 + F_2 C_2 - F_1 C_1 = 0$$

- Energy balance,

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

- Heat transfer [one of the following equations]

$$Q_{desorber} = F_3(h_3 - h_4)$$

$$LMTD = \frac{(T_4 - T_6) - (T_3 - T_5)}{ln[\frac{T_4 - T_6}{T_3 - T_5}]}$$

$CAT = (T_4 - T_6)$ *For IPINCH = -1 (temperature pinch at cold side)*

$CAT = (T_3 - T_5)$ *For IPINCH = 1 (temperature pinch at hot side)*

$EFF = \frac{T_3 - T_4}{T_3 - T_6}$ *For IPINCH = -1 (temperature pinch at cold side)*

$EFF = \frac{T_5 - T_6}{T_3 - T_6}$ *For IPINCH = 1 (temperature pinch at hot side)*

- Mass transfer – liquid side

$$T_5 = T_{5E}(P_5, C_5) + DEVL = 0$$

- Heat/mass transfer – vapor side

$T_2 = T_5 + DEVG$ *For type 1 and 2*

$T_2 = T_6 + DEVG$ *For type 3*

- Equilibrium at point 2

$$C_2 = C_{2E}(P_2, T_2)$$

- Adiabatic equilibrium at point 6

$$\text{If} \quad T_6 = T_1, \text{ then } C_6 = C_1 \text{ and } F_6 = F_1$$

$$\frac{h_1 - h_2}{h_1 - h_6} = \frac{C_1 - C_2}{C_1 - C_6}$$

$$T_6 = T_{6E}(P_6, C_6) = 0$$

$$\frac{F_6}{F_1} = \frac{C_1 - C_2}{C_6 - C_2}$$

The constraints are:

$$T_3 > T_5$$

$$T_4 > T_6$$

$$T_5 > T_6$$

$$T_3 \geq T_4$$

4.1.3. Heat Exchanger

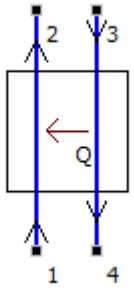

Fig. 4-3 Heat Exchanger

The heat exchanger is referred to in the code as component type 3 and is described schematically in figure 4-3.

A hot stream that enters at state 3 and leaves at state 4 transfers heat to a cold stream that enters at state 1 and leaves 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 →4 in crossflow, co-flow or counter-flow to the cold stream 1 →2 (sub-type 1, 2 and 3, respectively). The governing equations are:

- Energy balance,

$$F_3(h_3 - h_4) = F_1(h_2 - h_1)$$

- Heat transfer [one of the following equations]

$$\mathrm{Q_{heatExchanger}} = \mathrm{F_1(h_2 - h_1)}$$

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

$\mathrm{CAT} = (\mathrm{T_4 - T_1})$ For IPINCH = -1 (temperature pinch at cold side)

$\mathrm{CAT} = (\mathrm{T_3 - T_2})$ For IPINCH = 1 (temperature pinch at hot side)

$\mathrm{EFF} = \frac{\mathrm{T_3 - T_4}}{\mathrm{T_3 - T_1}}$ For IPINCH = -1 (temperature pinch at cold side)

$EFF = \frac{T_2 - T_1}{T_3 - T_1}$ For IPINCH = 1 (temperature pinch at hot side)

The constraints are:

$$\mathrm{T_3 > T_2}$$

$$\mathrm{T_4 > T_1}$$

$$\mathrm{T_3 \geq T_4}$$

$$T_2 \geq T_1$$

4.1.4. Condenser

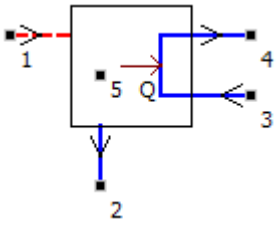

Fig. 4-4 Condenser

The condenser is referred to in the code as component type 4 and is described schematically in figure 4-4.

A stream of vapor that is either saturated or superheated at the condenser pressure enters the component at state 1. If it is superheated, it first cools to the condensing temperature at state 5 (point inside the component). 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 users.

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,

$$F_3(h_4 - h_3) = F_1(h_1 - h_2)$$

- Heat transfer [one of the following equations]

$$Q_{condenser} = F_3(h_4 - h_3)$$

$$LMTD = \frac{(T_5 - T_4) - (T_2 - T_3)}{\ln[\frac{T_5 - T_4}{T_2 - T_3}]}$$

$CAT = (T_2 - T_3)$ For IPINCH = -1 (temperature pinch at cold side)

$CAT = (T_5 - T_4)$ For IPINCH = 1 (temperature pinch at hot side)

$$\text{EFF} = \frac{T_5 - T_2}{T_5 - T_3} \text{ For IPINCH} = \text{-1 (temperature pinch at cold side)}$$

$$EFF = \frac{T_4 - T_3}{T_4 - T_3} \text{ For IPINCH} = 1 \text{ (temperature pinch at hot side)}$$

- Mass transfer

$$T_2 = T_{2E}(P_2, C_2) + DEVL = 0$$

- Heat/mass transfer – vapor side

$$T_2 = T_5 + DEVG \text{ } \textit{For type 1 and 2}$$

$$T_2 = T_6 + DEVG \text{ } \textit{For type 3}$$

- Equilibrium at point 5

$$T_5 = T_{5E}(P_5, C_5)$$

The constraints are:

$$T_5 > T_4$$

$$T_2 > T_3$$

$$T_4 \geq T_3$$

$$T_1 \geq T_5$$

$$T_5 \geq T_2$$

4.1.5. Evaporator

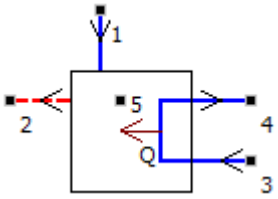

Fig. 4-5 Evaporator

The evaporator is referred to in the code as component type 5 and is described schematically in figure 4-5.

A stream of subcooled liquid or saturated vapor-liquid mixture at the evaporator pressure enters the component at state 1. If it is subcooled, it first heats to the evaporation temperature at state 5 (point inside the component). 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 users.

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,

$$F_3(h_3 - h_4) = F_1(h_2 - h_1)$$

- Heat transfer [one of the following equations]

$$Q_{evaporator} = F_3(h_3 - h_4)$$

$$LMTD = \frac{(T_3 - T_2) - (T_4 - T_5)}{\ln[\frac{T_3 - T_2}{T_4 - T_5}]}$$

$CAT = (T_4 - T_5)$ For IPINCH = -1 (temperature pinch at cold side)

$CAT = (T_3 - T_2)$ For IPINCH = 1 (temperature pinch at hot side)

$EFF = \frac{T_3 - T_4}{T_3 - T_5}$ For IPINCH = -1 (temperature pinch at cold side)

$EFF = \frac{T_2 - T_5}{T_3 - T_5}$ For IPINCH = 1 (temperature pinch at hot side)

- Equilibrium at point 2

  (When $T_5 \neq T_2$)

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

- Equilibrium at point 5

  (When $T_5 \neq T_1$)

$$T_5 = T_{5E}(P_5, C_5)$$

The constraints are:

$$T_3 > T_2$$

$$T_4 > T_5$$

$$T_3 \geq T_4$$

$$T_5 \geq T_1$$

$$T_2 \geq T_5$$

4.1.6. Valve

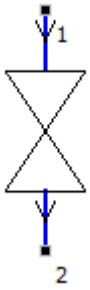

Fig. 4-6 Valve

The expansion valve is referred to in the code as component type 6 and is described schematically in figure 4-6.

As stream of 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 users 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,

$$h_1 - h_2 = 0$$

- Equilibrium at point 2,

$$\mathrm{T_2} = \mathrm{T_{2E}}(\mathrm{P_2}, \mathrm{C_2})$$

- Flow rate – pressure difference relation (for sub-type 2 throttle valve only)

$$\mathrm{F_1} = \mathrm{C_{vlv}}(\mathrm{P_1} - \mathrm{P_2})^{\mathrm{p}}$$

where $\mathrm{C_{vlv}}$ is the valve coefficient and p is a power law coefficient

- Thermostatic operation (for sub-type 3 thermostatic valve only)

$$\mathrm{T_3} - \mathrm{T_2} = \Delta\mathrm{T}$$

where, $\Delta\mathrm{T}$ is the temperature difference to be maintained as fixed

4.1.7. Mixer

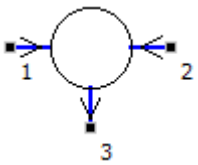

Fig. 4-7 Mixer

The mixer is referred to in the code as component type 7 and is described schematically in figure 4-7.

Two streams (states 1 and 2) may be at different temperatures, flow rates, concentrations, and vapor fractions but enter the component 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 180ºor at 90ºto each other (sub-type 1 and 2, respectively). The governing equations are:

- Overall mass balance,

$$F_1 + F_2 - F_3 = 0$$

- Absorbent/absorbate mass balance,

$$F_1 C_1 + F_2 C_2 - F_3 C_3 = 0$$

- Energy balance,

$$F_1 h_1 + F_2 h_2 - F_3 h_3 = 0$$

4.1.8. Splitter

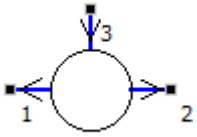

Fig. 4-8 Splitter

The splitter is referred to in the code as component type 8 and is described schematically in figure 4-8.

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 180º or at 90ºto each other (sub-type 1 and 2, respectively). The governing equations are:

- Overall mass balance,

$$F_1 + F_2 - F_3 = 0$$

- Split ratio (when specified),

$$F_1 = sF_3$$

where s is the split ratio.

4.1.9. Rectifier

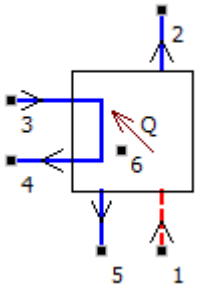

Fig. 4-9 Rectifier

The rectifier is referred to in the code as component type 9 and is described schematically in figure 4-9.

A stream of saturated or superheated vapor at state 1 enters the component at the rectifier pressure. If superheated, the vapor first cools down to equilibrium at state 6 (point inside the component). 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:

- Mass balance,

$$F_1 - F_2 - F_5 = 0$$

- Absorbent/absorbate mass balance

$$F_1C_1 - F_2C_2 - F_5C_5 = 0$$

- Energy balance,

$$F_3(h_4 - h_3) - (F_1h_1 - F_2h_2 - F_5h_5) = 0$$

- Heat transfer [one of the following equations]

$$Q_{rectifier} = F_3(h_4 - h_3)$$

$$LMTD = \frac{(T_6 - T_4) - (T_2 - T_3)}{ln[\frac{T_6 - T_4}{T_2 - T_3}]}$$

$CAT = (T_2 - T_3)$ *For IPINCH = -1 (temperature pinch at cold side)*

$CAT = (T_6 - T_4)$ *For IPINCH = 1 (temperature pinch at hot side)*

$EFF = \frac{T_6 - T_2}{T_6 - T_3}$ *For IPINCH = -1 (temperature pinch at cold side)*

$EFF = \frac{T_4 - T_3}{T_6 - T_3}$ *For IPINCH = 1 (temperature pinch at hot side)*

- Mass transfer – liquid side,

$$T_5 = T_{5E}(P_5, C_5) + DEVL$$

- Heat/mass transfer – vapor side,

$$T_6 = T_5 + DEVG$$

- Equilibrium at point 2

$$C_2 = C_{2E}(P_2, T_2)$$

- Equilibrium at point 6

  (When$T_6 \neq T_1$)

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

The constraints are:

$$T_6 > T_4$$
$$T_5 > T_4$$
$$T_2 > T_3$$
$$T_6 \geq T_5$$
$$T_4 \geq T_3$$
$$T_6 \geq T_2$$
$$T_1 \geq T_6$$

4.1.10. Analyser

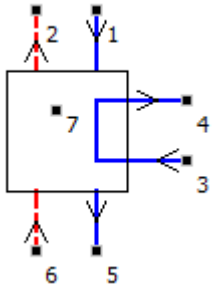

Fig. 4-10 Analyser

The analyser is referred to in the code as component type 10 and is described schematically in figure 4-10.

A stream of liquid solution entering at state 1, possibly superheated or subcooled, reaches equilibrium at state 7 internally (point inside the component) and leaves at state 5. It interacts with a stream of vapor entering at state 6 and leaving at 2, in counter-flow 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,

$$F_1 + F_6 - F_2 - F_5 = 0$$

- Absorbent/absorbate mass balance

$$F_1 C_1 + F_6 C_6 - F_2 C_2 - F_5 C_5 = 0$$

- Energy balance,

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

- Heat transfer (for sub-type 1 and 2) [one of the following equations]

$$Q_{analyser} = F_3(h_3 - h_4) \; \textit{For sub-type 1}$$

$$Q_{analyser} = F_3(h_4 - h_3) \; \textit{For sub-type 2}$$

$$LMTD = \frac{(T_4 - T_7) - (T_3 - T_5)}{ln[\frac{T_4 - T_7}{T_3 - T_5}]}$$

$CAT = (T_4 - T_7)$ *For IPINCH = -1 (temperature pinch at cold side) and*

$CAT = (T_3 - T_5)$ *For IPINCH = 1 (temperature pinch at hot side), for sub-type 1;*

$CAT = (T_5 - T_3)$ *For IPINCH = -1 (temperature pinch at cold side) and*

$CAT = (T_7 - T_4)$ *For IPINCH = 1 (temperature pinch at hot side), for sub-type 2;*

$EFF = \frac{T_3 - T_4}{T_3 - T_7}$ *For IPINCH = -1 (temperature pinch at cold side) and*

$EFF = \frac{T_5 - T_7}{T_3 - T_7}$ *For IPINCH = 1 (temperature pinch at hot side), for sub-type 1;*

$EFF = \frac{T_7 - T_5}{T_7 - T_3}$ *For IPINCH = -1 (temperature pinch at cold side) and*

$EFF = \frac{T_4 - T_3}{T_7 - T_3}$ *For IPINCH = 1 (temperature pinch at hot side), for sub-type 2;*

- Mass transfer – liquid side,

$$T_5 = T_{5E}(P_5, C_5) + DEVL$$

- Heat/mass transfer – vapor side,

$$T_2 = T_7 + DEVG$$

- Equilibrium at point 2

$$C_2 = C_{2E}(P_2, T_2)$$

- Adiabatic equilibrium at point 7

If $T_7 = T_1$, then $C_7 = C_1$ and $F_7 = F_1$

$$\frac{h_1 - h_2}{h_1 - h_7} = \frac{C_1 - C_2}{C_1 - C_{76}}$$

$$T_7 = T_{7E}(P_7, C_7) = 0$$

$$\frac{F_7}{F_1} = \frac{C_1 - C_2}{C_7 - C_2}$$

The constraints are:

$$T_3 \geq T_4$$

$$T_4 > T_7$$

$$T_3 > T_5$$

$$T_5 > T_7$$

For sub-type 1

$$T_4 \geq T_3$$

$$T_7 > T_4$$

$$T_5 > T_3$$

$$T_7 > T_5$$

For sub-type 2

4.1.11. Compressor

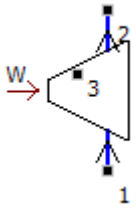

Fig. 4-11 Compressor

The compressor is referred to in the code as component type 11 and is described schematically in figure 4-11.

A stream of gas at low pressure enters at state 1 and is compressed to a high pressure (state 2). State 3 (point inside the component) indicates the condition the gas would attain under isentropic compression. The governing equations are:

- Isentropic process,

$$S_1 - S_3 = 0$$

- Deviation from the ideal process (when specified)

$$(h_3 - h_1) - e(h_2 - h_1) = 0$$

where e is the isentropic efficiency.

4.1.12. Pump

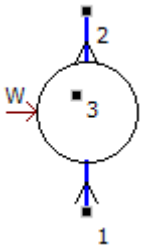

Fig. 4-12 Pump

The pump is referred to in the code as component type 12 and is described schematically in figure 4-12.

A stream of liquid at low pressure enters at state 1, and is pumped to a high pressure (state 2).

State 3 (point inside the component) indicates the condition the liquid would attain under isentropic pumping. The governing equations are:

- Isentropic process,

$$(h_3 - h_1) - \frac{P_3 - P_1}{\rho_1} = 0$$

where ρ_1 is the liquid density.

- Deviation from the ideal process (when specified)

$$(h_3 - h_1) - e(h_2 - h_1) = 0$$

where e is the pumping efficiency.

4.2.LIQUID DESICCANT COMPONENTS

4.2.1. Adiabatic components using effectiveness model

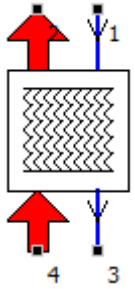

Fig. 4-13 Adiabatic dehumidifier/regenerator using effectiveness model

The adiabatic components using effectiveness model include adiabatic dehumidifier and adiabatic regenerator that are calculated using effectiveness model. The dehumidifier is referred to as type 16 (adiabatic dehumidifier) sub-type 4, and the regenerator is referred to as component type 18 (adiabatic regenerator) sub-type 4. The figure shows the shared schematic of both components.

A stream of solution enters at state 1 from top of the device and contact with a stream of air entering from the bottom at state 4. For dehumidifier, the entering solution is usually cool and dense, and in contact with air it condenses the moisture in the air to transfer into the solution. The air is thus dehumidified and cooled when leaving at state 2, while the solution become warm and weak leaving at state 3. For regenerator, the entering solution is usually hot and dilute, and in contact with air it releases water to evaporate into the air. The solution thus becomes dense and cool leaving at state 3, while the air becomes humid and warm leaving at state 2.

The governing equations are:

- Energy balance for the entire device:

$$F_1 \cdot h_1 - F_3 \cdot h_3 + F_4 \cdot (h_4 - h_2) = 0$$

- Mass balance of desiccant salt for the entire device:

$$F_1 C_1 - F_3 C_3 = 0$$

- Moisture mass balance for the entire device:

$$F_3 - F_1 - F_4 \cdot (w_4 - w_2) = 0$$

- air outlet enthalpy:

$$h_2 = h_4 - \varepsilon[h_4 - h_{1E}(T_1, C_1)]$$

where ε is the effectiveness of the device

- air outlet humidity ratio:

$$w_2 = w_{eff}(h_4, h_2, e) + [w_4 - w_{eff}(h_4, h_2, \varepsilon)] \cdot exp[-NTU(\varepsilon, F_1, F_4)]$$

Where, w_{eff} is the effective constant equilibrium humidity ratio at solution surface, NTU is the number of transfer unit of the device

4.2.2. Adiabatic components using finite difference model

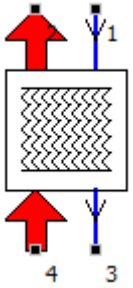

Fig. 4-14 Adiabatic dehumidifier/regenerator using finite difference model

The adiabatic components using finite difference model include adiabatic dehumidifier and adiabatic regenerator that are calculated using finite difference model. The dehumidifier is referred to as component type 16 (adiabatic dehumidifier) sub-type 1-3, and the regenerator is referred to as component type 18 (adiabatic regenerator) sub-type 1-3. The figure shows the schematic of counter-flow configuration of both types of components.

A stream of solution enters at state 1 from top of the device and contact with a stream of air entering from the bottom at state 4. For dehumidifier, the entering solution is usually cool and dense, and in contact with air it condenses the moisture in the air to transfer into the solution. The air is thus dehumidified and cooled when leaving at state 2, while the solution become warm and

weak leaving at state 3. For regenerator, the entering solution is usually hot and dilute, and in contact with air it releases water to evaporate into the air. The solution thus becomes dense and cool leaving at state 3, while the air becomes humid and warm leaving at state 2.

To apply finite difference method, the component is divided into a number of segments along its height (and also along the width for cross-flow configuration). For each segment control volume, conservation equations and transfer equations are established. The parameters of each segment are "integrated" with fixed inputs and initial guess values from the top (state 1 and 2), till it reaches the bottom of the components. The results of the "integral" process are compared with fixed inputs at the bottom (state 3 and 4), and the solver will adjust guess values towards convergence.

The governing equations for each segment are:

- Energy balance for the control volume

$$F_s \cdot \frac{dh_s}{dZ} + h_s \cdot \frac{dF_s}{dZ} - F_a \cdot \frac{dh_a}{dZ} = 0$$

Where, dZ is the height of each segment, dZ = Z/n, Z is the total height of the component, n is the number of segments

- Mass balance for desiccant salt

$$F_s \cdot C_s - \left(F_s + \frac{dF_s}{dZ}\right) \cdot \left(C_s + \frac{dC_s}{dZ}\right) = 0$$

- Mass balance for moisture balance

$$\frac{dF_s}{dZ} - F_a \cdot \frac{dw_a}{dZ} = 0$$

- The change of air humidity ratio in the segment is:

$$\frac{dw_a}{dZ} = \frac{NTU_m}{Z}[w_a - w_{eq}(T_s, C_s)]$$

where NTU_m is the number of transfer unit of the entire component

- The change of air enthalpy in the segment is:

$$\frac{dh_a}{dZ} = \frac{NTU_m \cdot Le}{Z} \cdot [h_a - h_{eq}(T_s, C_s) + (\frac{1}{Le} - 1) \cdot r \cdot (w_a - w_{eq}(T_s, C_s))]$$

Where Le is the Lewis number of air, r is the evaporation/condensation enthalpy of water

Given the NTU_m ,Le and the boundary conditions, the following solving process can be applied in each iteration:

1> Assume a set of initial guess values for air outlet: t_2 , w_2 ;

2> Calculate the enthalpy h_{eq} and humidity ratio w_{eq} of air in equilibrium at the solution surface with temperature T_s and concentration C_s;

3> Calculate the air enthalpy difference dh_a;

4> Add the air enthalpy difference to current air enthalpy for next control volume;

5> Calculate the air humidity ratio difference dw_a;

6> Add the air humidity ratio difference to current one for next control volume;

7> Calculate the next control volume air temperature using $T_a = \frac{(h_a - r \cdot w_a)}{C_{p,a} + C_{p,v} \cdot w_a}$;

8> Calculate the concentration difference;

9> Calculate next control volume C_s and F_s;

10> Calculate solution temperature difference using $dT_s = \frac{F_a \cdot dh_a}{F_s \cdot C_{p,s}}$;

11> Calculate next control volume T_s.

4.2.3. Internally cooled/heated components using finite difference model

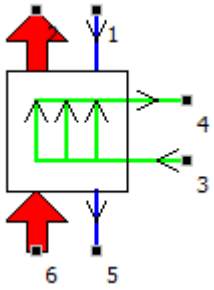

Fig. 4-15 internally cooled dehumidifier/internally heated regenerator model

The internally cooled/heated components include internally cooled dehumidifier and internally heated regenerator, and both types of components are calculated using finite difference model. The dehumidifier is referred to as component type 17 (adiabatic dehumidifier) sub-type 1-9, and the regenerator is referred to as component type 19 (adiabatic regenerator) sub-type 1-9. The figure shows and example schematic of both types of components.

A stream of solution enters at state 1 from top of the device and contact with a stream of air entering from the bottom at state 4. For dehumidifier, the entering solution is usually cool and dense, and in contact with air it condenses the moisture in the air to transfer into the solution. The

air is thus dehumidified and cooled when leaving at state 2, while the solution become warm and weak leaving at state 5. For regenerator, the entering solution is usually hot and dilute, and in contact with air it releases water to evaporate into the air. The solution thus becomes dense and cool leaving at state 5, while the air becomes humid and warm leaving at state 2. A stream of cooling/heating fluid enters the component at state 3, and it transfers heat with the solution (as well as the air if the surface is not entirely wetted). For dehumidifiers the heat transfer fluid is usually cold to take away the condensation heat of moisture from the solution to keep it cool. For regenerators the heat transfer fluid is usually hot to compensate the evaporation heat of water to the solution to keep it warm.

To apply finite difference method, the component is divided into a number of segments along its height (and also along the width for cross-flow configuration). For each segment control volume, conservation equations and transfer equations are established. The parameters of each segment are "integrated" with fixed inputs and initial guess values from the top (state 1, 2, and 4), till it reaches the bottom of the components. The results of the "integral" process are compared with fixed inputs at the bottom (state 3, 5, and 6), and the solver will adjust guess values towards convergence.

The governing equations for each segment are:

- Energy conservation of the control volume

$$\frac{d(F_s h_s)}{dZ} - \frac{F_a dh_a}{dZ} - \frac{F_w Cp_w dT_w}{dZ} = 0$$

- Mass conservation of the control volume

$$\frac{dF_s}{dZ} - \frac{F_a dw_a}{dZ} = 0$$

- Salt mass conservation of the control volume

$$\frac{d(F_s C_s)}{dZ} = 0$$

- The change of air humidity ratio in the segment is:

$$\frac{dw_a}{dZ} = \frac{NTU_m}{Z}[w_a - w_{eq}(T_s, C_s)] \cdot \beta$$

where NTU_m is the number of transfer unit between air and solution in the entire component defined as $NTU_m = h_D A/F_a$, and β is the wetness level defined as the ratio of wetted area over total area

- The change of air enthalpy in the segment is:

$$\frac{dh_a}{dZ} = \frac{NTU_m \cdot Le \cdot \beta}{Z} \cdot \left[h_a - h_{eq}(T_s, C_s) + \left(\frac{1}{Le} - 1\right) \cdot r \cdot \left(w_a - w_{eq}(T_s, C_s)\right)\right] + NTU_a(T_w - T_a)(1 - \beta)$$

Where Le is the Lewis number of air, r is the evaporation/condensation enthalpy of water, NTU_a is the number of heat transfer unit between air and the heat transfer fluid defined as $NTU_a = \frac{h_A A}{C_{p,w} F_w}$

- Heat transfer water –solution

$$\frac{dT_w}{dZ} = NTU_s(T_w - T_s) \cdot \beta + NTU_a(T_w - T_a)(1 - \beta)$$

where NTU_s is the number of heat transfer unit between solution and the heat transfer fluid defined as $NTU_s = \frac{h_W A}{C_{p,w} F_w}$

Given NTU_s, NTU_m, NTU_a, Le and inlet conditions (if to iterate from top (solution inlet), T_1, F_1, C_1, T_2, w_2, F_2, T_4, T_4), the solving process is:

12> Calculate equilibrium humidity ratio $w_{eq}(T_s, C_s)$ and enthalpy $h_{eq}(T_s, w_{eq})$;

13> Calculate air enthalpy change dh_a, then next step h_a;

14> Calculate humidity ratio change dw_a, then next step w_a;

15> Use next step h_a and w_a to calculate next step T_a;

16> Calculate water temperature change dT_w, then next T_w;

17> Calculate dF_s, then next F_s;

18> Calculate dC_s, then next C_s;

19> Calculate dh_s, then next h_s;

20> Calculate T_s with next h_s and C_s;

The output of each iteration is the outlet conditions: F_5, T_5, C_5, T_6, w_6, T_3. These values will be compared with input values in the solver, and adjustment of guess values will be made for next step iteration towards convergence.

5. USING THE GRAPHICAL USER INTERFACE

The graphical user interface has been described briefly in Section 2. In this section, detailed descriptions of every function provided in the SorpSim menu are presented.

5.1.MAIN WINDOW

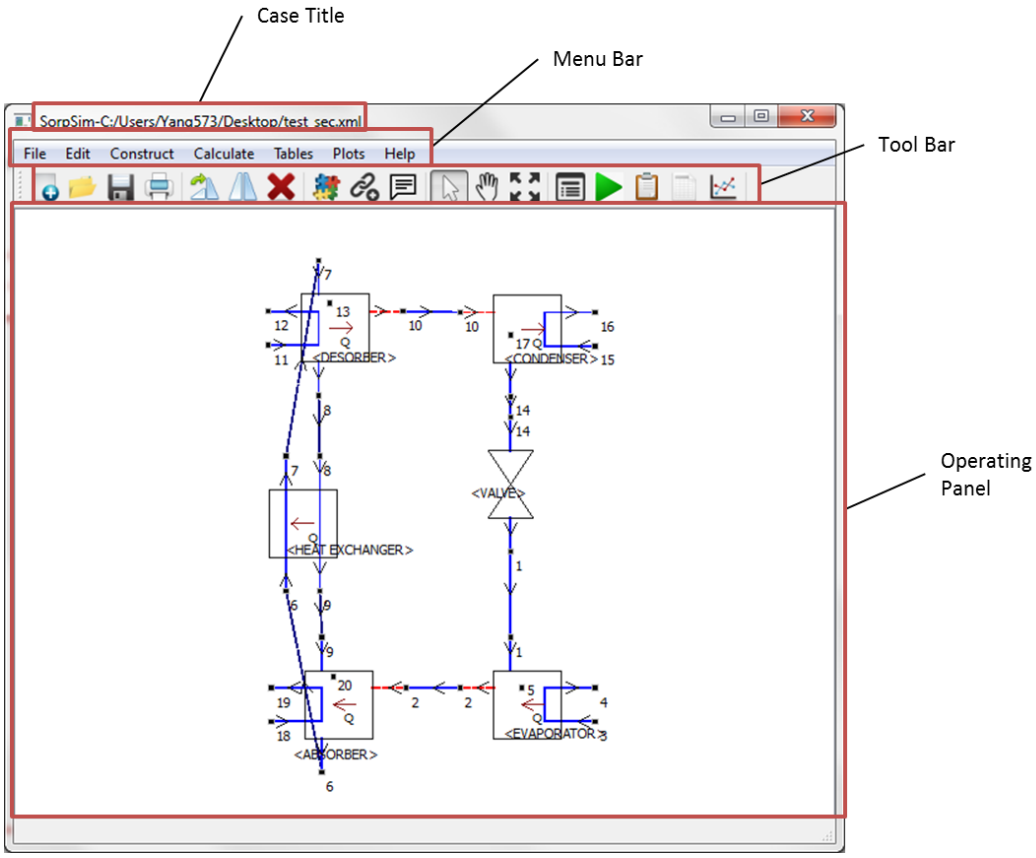

Fig. 5-1 Main window of the SorpSim GUI

The Main Window is the major interface window that the SorpSim software is running on. As shown in the figure 5-1, the Main Window is consisted of several parts. The case title indicates the name and directory of current case. The menu bar contains access to all the functions provided by the software. Under each menu item there are a series of buttons to trigger different functions such as saving/loading cases, editing case unit system, adding a new component, or starting a new parametric table to further analyze a calculated case. The tool bar located under the menu bar provides quick access to some of the most frequently used functions for the convenience of users. These functions include save/load, delete selected item, add link/text, zoom to fit, calculate, and initiate table/plot window. The operating panel is where the cycle diagram is constructed and edited with components and links.

5.2.FILE MENU

The file menu contains functions related to the cases file and printing/exporting the cycle diagram.

- New—initiates a new work session. All components, variables, tables, and plots will be cleared along with the program being reconfigured to its default settings. If an unsaved case is present, users will be prompted to save the case. Then users are asked to decide the unit system and working fluids that will be used in the new case.

- Open—allows users to resume to an existing case. If current case is not saved, the program will first ask users to save the current case. Then users need to locate the existing case file in the dialog.

- Recent File— provides a quick access to the 5 most recent cases that have been opened or saved by the SorpSim software. The list of recent files is updated every time users save or loads a case file.

- Import .out File—provides the direct compatibility to the cases that were constructed and calculated using the ABSIM software. This function loads the result case files (with suffix .out) generated by the ABSIM software, and restores all the cycle system data into a new case in the SorpSim software. Since .out file doesn't contain location information for each component, all the components are placed in a row. The cycle diagram can be better arranged by dragging components and rotating/flipping them.

- Save—saves all information relating to the case in an .xml file on the disk using the file name for which it was last saved. The case can be later restored with the Open function..

- Save as—saves all information relating to the case in an .xml file on the disk using the file name for which it was last saved. The function always asks users to specify the directory and file name for saving the case. And if the specified file name and directory overlaps with existing files, the program will ask users whether to overwrite the existing file.

- Print—allows users to print the cycle diagram as it is shown on the operating panel. It will evoke a preview dialog as shown in the figure. The native printer manager of current operating system will handle the printing task then.

- Export to File—allows users to export the cycle diagram as it is shown on the operating panel into an image (.png) file or a PDF file. Users are asked to specify the file name and directory for the new file.

- Exit—provides an elegant way of closing the software. If current case is not saved, the program will first ask users to save the case before the program is closed.

5.3.EDIT MENU

The Edit menu contains functions related to making changes to the system cycle, both to the configuration and to the parameters. Also the Edit menu contains functions to adjust operating mode and control the view of the operating panel.

- Delete –removes the selected item on the operating panel, either a component, a custom text item, or a link, from the system cycle.

- Properties—evoke a dialog for users to edit the properties of the currently selected item on the operating panel. Different dialogs are evoked for different types of components, and a special dialog is evoked for state points. Users can also access the properties dialogs of components and state points by double-clicking them on the operating panel.

- Master Control Panel—provides a system-level overview and edit platform for users to review and control each parameter in the cycle setup. The function evokes a dialog with two tabs of control tables corresponding to all the components and all the state points in the system cycle. In the control tables, users can review and make changes to all the parameters in the cycle. In the control table for state points, users can toggle the parameters for each state point between unknown and fixed as input, and enter values for those parameters that are defined as "fixed inputs".
    - The "Export" button allows users to print the current control table as it is. By clicking the button, the native operating system's printer handler will be evoked.
    - The "Set Guess Value" button will evoke a dialog with all the values for each state point parameter listed in a table. Than greyed cells indicate those parameters are fixed as input and cannot be edited. The white cells contain the initial guess value for that parameter, and users can adjust the values closer to the final results for quicker convergence.
    - The "Update" button on the Guess Value dialog will update all the guess values in the system with the results from the previous calculation. The "Export" button on the Guess Value dialog allows users to either copy the guess value table to the clip board so that it can be pasted into spreadsheet, or print the table as it is via the operating system's printer handler.

- By clicking the “Apply” button on the Guess Value dialog the guess values in the cycle will be overwritten with the values in the table. Otherwise if users close the dialog with the “x” button or cancel, no change to the guess values will be enforced.

- System Settings—provides users with the capability to switch unit systems and change working fluid inventory for the current case. For each unit system, different units are provided for temperature, pressure and energy.

- Rotate/Flip—allows users to transform the currently selected component on the operating panel so that its links will not be tangled with other links. Each time the rotate button is triggered, it will rotate the component clockwise for 90 degree. The horizontal flip button and the vertical flip button flip the component horizontally and vertically.

- Select/Pan—is corresponding to different operating mode of the mouse action on the operating panel: the select mode allows selecting and dragging items on the panel, while the pan mode allows users to move the entire cycle diagram on the panel. The two buttons can be toggled, namely only one of the two modes is activated at one time.

- Zoom to fit—provides an easy one-click approach for users to show the entire cycle on the panel no matter how large the cycle diagram may be. It adjusts the scale of the cycle diagram, and centers the panel at the mean coordinate of all the components. Users can also scale up and down on the operating panel by using the mouse wheel.

5.4.CONSTRUCT MENU

The construct menu contains functions to add items into the system cycle.

- Component—opens a dialog for user to add a new component into the system cycle. The component dialog contains the inventory of all the available components provided by the software in a tree structure. Some of the components contain different sub-types by configuration. Once a component is selected in the tree, its preview is displayed on the right panel, along with a short descriptive text to explain the state points and stream layout in the component. By clicking “Select” button, users confirm the new component and then can place the new component onto the operating panel. The indexes of the component and all of its state points are automatically assigned. Otherwise if users close the dialog or clicks “Cancel”, the new component will be canceled, and no change will be made to the system cycle.

- Link—allows users to make links between state points to form streams. The linked two state points are considered as one state point and share the same parameters. Once the linking procedure begins, users will select two state points on the operating panel in sequence by double-clicking them, and then define the shared data for the combined point. If users would like to terminate the linking procedure, he can press the escape button “ESC” to exit the function.

- Text—allows users to add custom text items onto the operating panel. The function evokes a dialog for users to define the text item. Users can enter the content of the text item, set the size, color, and style of the item before adding it onto the operating panel. Text items already on the operating panel can be edited by a double-clicking action on the item. All the text items that are added onto the operating panel are stored in the case file and will be restored when the case is loaded from file.

5.5.CALCULATE MENU

The calculate menu contains functions to control the simulation procedure, as well as functions to review calculation details and results.

- Additional Equations—allows users to add additional equal-correlation to those state point parameters that are identified as “unknown”. One example for such correlation is where users would like the desorber solution outlet and vapor outlet to have the same temperature. Since such correlations are problem-specific, they are managed using the Additional Equations function according to users’ need. For each type of parameter, the unknowns that share the same value are indicated in the same group. The correlations are managed by adding or removing parameters in the group. In the dialog shown above, users can add a new correlation by selecting state points and press “add new equation” button, or modify and remove the correlations by selecting one of the existing groups and press the corresponding button.

- Run—helps users to setup the control for the simulation procedure, and starts calculating the current case. The calculation control dialog will first be evoked where users specify the iteration limit and tolerance for convergence. Once users press the “ok” button, the calculation is initiated, and the report message is shown after the calculation finishes. If calculation is completed without any error, the report will display the above message. If the calculation is terminated before convergence is reached due to some problems, the problems will be described in the report message. The reporting dialog provides quick access to result display and convergence details functions, which will be described in the following sections.

- Convergence Details—provides users with insight of the previous calculation in the hope that it will help users to review the calculation process and reach successful calculation.
    - The “Function Residuals” tab contains all the equations that were used in the previous calculation along with their final residuals. The table contains the components in which each equation is included, the equation names (some with specific form of the equation as well), and the residual values. By selecting one equation and pressing the “Locate Component” button, the program will jump to the operating panel and highlight the component with the selected equation.

- The “Guess Values” tab of the convergence detail dialog contains the guess values that were used in the last iteration of calculation. Since some of the state points have the equal-correlations and share some parameter values, thus the table presents the guess values with the state point indexes, variable type, and the last guess value used in the iteration.

- Show Results—superimposes the calculation results onto the operating panel. The result items are initialized beside the corresponding state points and components. Users can drag the items around to arrange a better presentation. The coordinates of the result items are stored in the case file and will be retrieved when the case is loaded from the file.

- Choose Results—allows users to choose the parameters to display on the operating panel. Results of component, state point, and the system level parameters can be chosen to display. After the “Apply” button is clicked, the displayed result items will be updated with the selected results.

- Results Table—displays all the calculation results for components and state points in tabular form. For the state point tab, the green-lighted cells indicate those parameters are set as fixed input in the calculation. The white-lighted cells are the results of calculation. The components tab contains the results of calculated component parameters. The content of the results table can be copied into clipboard with or without header via the “export” button.

5.6.TABLES MENU

The Tables Menu contains functions to generate new parametric tables, as well as to access existing tables.

- New Parametric Table—allows users to generate a new parametric table based on current system cycle. Users are asked about the title of the table, number of runs, as well as the input and output parameters of the table. The inputs of the table are selected from the heat transfer value of each component under current heat transfer method, and the state point parameters that are currently set as "fixed input". By pressing the "add input" button, the program jumps to the operating panel, where users can double-click a component or a state point to select the parameter to add to the table inputs. The outputs of the table are selected from all the calculated results of component and state point parameters, as well as the system COP and the system capacity if the heat duties of some components are specified as COP numerator and denominator. The selection procedure of the table output parameters are the same as the inputs, with system COP and capacity being added with a press on the corresponding button. After setting up at least one input and one output, users can press the OK button to initiate the new parametric table.

- Table Window—opens up the table dialog that contains all the existing parametric tables under tabs. For each table the input parameters are displayed in black font, while the outputs are displayed in blue font. There are two rows of function buttons for editing the table values as well as making changes to the current table as a whole.
    - The "Insert Value" button will evoke a dialog where users can quickly insert a series of values for each input parameter. Users can switch between input parameters using the drop-down menu, choose to enter values or clear values, specify the range of rows to operate on, and enter the values to insert with the first value and patterns including linear (given last value), log (given last value), multiplier, and incremental.
    - The "Alter Runs" button will evoke a dialog where users can make changes to the number of runs. Users can either insert or remove runs at specific location in the table, with the number of inserted or removed runs entered by users.

- The “Edit Column” button will evoke a dialog where users can add or remove input and output parameters of the current table. The procedure of selecting input and output parameters is the same as when a new parametric table is generated.
- The “Calculate” button will initiate the calculation of the table using current global settings. If calculation is terminated at one row, the program will stop and indicate the problematic row with red highlight.
- The “Copy Table” button will make a copy of the current table and add it as a new tab into the table dialog.
- The “Delete Table” button will remove the current table along with all its data.
- The “Export” button provides a series of functions to export the current table content. “Copy to Clipboard” button will copy the table content with headers to the clipboard, which can be pasted into spreadsheet applications later. “Plot Table Results” button will generate a new parametric plot based on the current table. The plot function will be explained with detail in the following section. “Print Table” will evoke the native operating system’s printer handler and print the current table as it is. “Export Table to Text File” button allows users to export the content of the table with headers into a text file.

5.7.PLOTS MENU

The Plots Menu contains functions to generate new parametric and property plots, as well as to access existing plots.

- New Parametric Plot—allows users to generate a new parametric plot based on an existing parametric table. The function evokes a dialog as shown above. Users need to specify the title of the new plot, and select the data source from one of the existing parametric tables. Then users need to select one parameter for the x-axis and one for the y-axis. The new parametric plot will be initiated in the plot dialog evoked later once users press the OK button.

- New Property Plot—allows users to generate a new property chart as a reference to analyze current system cycle. Currently the program only provides Duhring Chart of LiBr-H2O working pairs. Users only need to specify a title for the new property plot and press OK to initiate the new plot.

- Plot Window—opens up the dialog that contains all existing plots (including parametric plots and property plots). Pressing the left mouse key on the plot will capture the current point coordinate and its status parameters. Right clicking on the plot will evoke a menu of options.
    - The "Zoom" button is a toggle button to turn on and off the zooming function. Once the zoom function is activated, press-and-drag mouse action on the plot will draw a rectangle, and releasing the mouse will lead to the plot zooming at the rectangular area. The zoom can be resumed by turning off the zoom function.
    - The "Format Plot" button will open a dialog to setup all the properties of the current plot.
        - The "General" tab in the format plot dialog contains general information about the current plot. If the plot is a property plot, the tab looks like the left one in the figure, containing the plot title, titles for both X and Y axis, the background color, as well as the background lines. The background lines are concentration lines for reference, and users can turn on the lines of interest or turn off the irrelevant lines to adjust the presentation on the plot. If the plot is a parametric plot, the tab looks like the right one in the

figure. There is no “background lines” in parametric plots, while the rest are the same under the “General” tab.

- The “Plot Options” tab contains options for each curve that users generated. Users can show and hide the selected curve by toggling on and off it in the curve list. Users can also edit or delete the current selected curve (the edit curve is similar to the overlay function for property plots, which will be introduced in the following sections). The curve title, width, color, and line style can be changed as well.
- The “Legend” tab contains controls of legend items in the plot. Users can turn on the legend by checking “enable legend” box. Then for external legend (which appears outside the plot region) or the internal legend (which is superimposed within the plot region), users can edit the orientation and position.
- The “Grid” tab contains controls of grid lines on the current plot. Users can choose whether to show X grid lines and Y grid lines on both major and minor scale. The line width, color, and line style can be defined as well.

- The “Overlay” button for property plot in the plot dialog. The button will evoke a dialog as shown in the figure above to add a new curve based on the current system cycle onto the plot. Users need to enter a name for the new curve, and then list all the involved state points in the loop in sequence. The state points can be selected one by one from the system diagram by pressing “Start selecting from cycle” button, or added by entering the index of the state point and pressing “Add by state point index” button. The order of the state points can be arranged using “Move up”, “Move down”, and “Remove” buttons. The program also provides a loop detection feature to automatically capture closed loops in the system for users to add a loop as a whole. By pressing “OK” button, the selected state points will be superimposed onto the plot in the re-arranged order. The existing curves on a property plot can be edited via the “Edit Curve” button in the “Plot Options” tab within the “Format Plot” function described above.
- The “Data Selection” button for parametric plot in the plot dialog. The button will evoke a dialog to re-select the X-axis and Y-axis parameters from the parametric table that the current plot is generated.

5.8.HELP MENU

The Help Menu contains functions to provide information about program functions, component models, and the general software. It also contains access to several pre-developed example cases.

- Help content—starts up the SorpSim help dialog. Users can find information about functions in the main menu, information about models for components, and other useful resources of the SorpSim software.

- The About SorpSim—displays information of the current SorpSim version and information of the developer and licenses.

- The example cases are pre-developed cases that can be used as the bases of developing variations of systems. These example cases for absorption systems are developed from example cases of the ABSIM software, and the desiccant example cases are developed by the SorpSim team.

NOMENCLATURE

A = heat transfer area (m^2 or ft^2)
C = absorbent/absorbate concentration (wt %)
C_p = specific heat (kJ/kg°K or Btu/lb°F)
CAT = closest approach temperature (°C or °F)
DEV = extent of deviation from equilibrium state (°C or °F)
DEVG = deviation from equilibrium, vapor side (°C or °F)
DEVL = deviation from equilibrium, liquid side (°C or °F)
EFF = heat transfer effectiveness (dimensionless)
F = mass flow rate (kg/s or lb/min)
h = specific enthalpy (kJ/kg or Btu/lb)
LMTD = logarithmic mean temperature difference (°C or °F)
Le = Lewis number of air
m = number of equations
NTU = number of transfer unit
n = number of unknowns
P = pressure (kPa or psia)
Q = heat transfer rate (kW or Btu/min)
r = evaporation enthalpy of water
S = entropy (kJ/kg K or Btu/lb°R)
T = temperature (°C or °F)
U = overall heat transfer coefficient (kW/m2°C or Btu/ft2°Fmin)
UA = overall heat transfer coefficient times area (kW/°C or Btu/°Fmin)
W = vapor mass fraction (dimensionless)
w = humidity ratio of moist air (kg/kg d.a. or lb/lb d.a.)
Z = height of the device
ρ = density (kg/m3or lb/ft3)

Subscripts

a = air, heat transfer between air and the heat transfer fluid
E = equilibrium
eff = assumed "effective" state that delivers the same outlet results
eq = equilibrium with the solution surface
i, j = state point index (numerical values may be used)
k = number of current iteration
L = liquid
m = mass transfer between air and solution
s = solution, heat transfer between solution and the heat transfer fluid
V,v = vapor
w = water/heat transfer fluid

APPENDIX A: DEVELOPMENT TOOLS

The SorpSim software is extensively developed using the Qt/C++ framework. All the graphical interface, functions and features are implemented within the Qt Creator in C++ language. Apart from using Qt's graphical libraries, Qwt, a 3rd party plugin library for plotting in Qt, and FABLE, a FORTRAN-C++ language conversion tool are used during development of SorpSim as well.

Qt/C++ Framework

Qt is a comprehensive C++ application development framework for creating cross-platform graphical user interface (GUI) applications. It supports compilation of a single source tree for applications that will run on Windows, Mac OS, Linux, and many other versions of UNIX with X11. The Qt toolkit uses standard C++ while also offering libraries and macros for graphic widget and GUI application development [15]. From 2005, programmers are allowed to create GNU Lesser General Public Licensed (LGPL) free/open source applications using Qt that dynamically link to the Qt libraries. This guarantees end users of applications developed on Qt-LGPL platform the freedom to use, study, share (copy), and modify the software [16]. Under this license, the SorpSim can be distributed to any end users free of charge, and allow them to modify the code with Qt.

The Qt Framework comes with extensive libraries containing classes for graphical widgets and objects such as dialogs and push buttons with methods for a wide range of operations. These libraries can be conveniently used for rapid development of even very complex user graphical interfaces, and SorpSim's multiple-dialog GUI along with much of their functions and features are deeply linked with these libraries. Qt also provides the library to parse and edit eXtensible Markup Language (XML) based files. The SorpSim's database to organize and store case data and information for parametric analysis is based on XML file, thus this aspect of Qt's capability was also exploited.

Finally, the cross-platform compilation feature of Qt makes it possible to develop on one platform (in our case on Windows), and compile the same code set on another computer platform (OS X for instance) to directly generate application for that platform. Only some minor platform-specific adjustment were to be made to adapt to the interface pattern of the new platform, and this "develop once, compile everywhere" feature greatly reduced the effort required to make the SorpSim available on the most popular computer platforms.

Qwt

The Qwt library (http://qwt.sf.net) contains GUI components and utility classes for generating scientific and technical plots on the Qt platform. The Qwt is a 3rd party plugin for Qt/C++ Framework, and it provides framework for 2D plots with data prepared in arrays. The Qwt is extensively used in SorpSim's parametric plot and property plot function. The Qwt library and included programs are provided under the terms of the GNU LGPL license, thus software using it can be distributed free of charge if not statically linked.

FABLE

Since the original source code of ABSIMW 5.0 that was provided to the SorpSim development team was written in FORTRAN, it was crucial that the code was converted into the same language that the GUI was developed. Fortunately such a tool had been developed by Grosse-Kunstleve RW et al [17] in the Lawrence Berkeley Laboratory. FABLE converts fixed-format FORTRAN sources to C++, and the generated C++ code is designed to be human-readable and suitable for further development. With the help of FABLE, the original source code of ABSIM scattered in 5 different FORTRAN files was converted and organized into one package written in C++ language. With minor adjustment of the converted code, the calculation engine along with the fluid and component libraries was ready to be integrated into the SorpSim GUI application.

APPENDIX B: EXAMPLE CYCLE CASES

This appendix contains cycle diagram for six sample cycles provided with the present version of SorpSim. The original case files can be found in the subdirectory \templates. However, try not to directly open and work on these files. Instead access these example files via the "template" or "example files" function and save the loaded case to some custom location and file name. The cycle diagrams are printed as drawn on the computer screen.

The following cycles are included:

- Absorption example cycles (developed from ABSIM [7] sample cycles)
  - SEC—Single-effect LiBr-water chiller
  - DECP—Double-effect LiBr-water chiller, parallel flow
  - DCCAP—Double-condenser-coupled, triple-effect LiBr-water chiller, parallel flow
  - GAX—Generator-absorber heat exchange, water-ammonia heat pump

- Liquid desiccant example cycles
  - LDAC—Liquid desiccant system using adiabatic dehumidifier and regenerator
  - LDAC2—Liquid desiccant system using internally cooled dehumidifier and internally heated regenerator

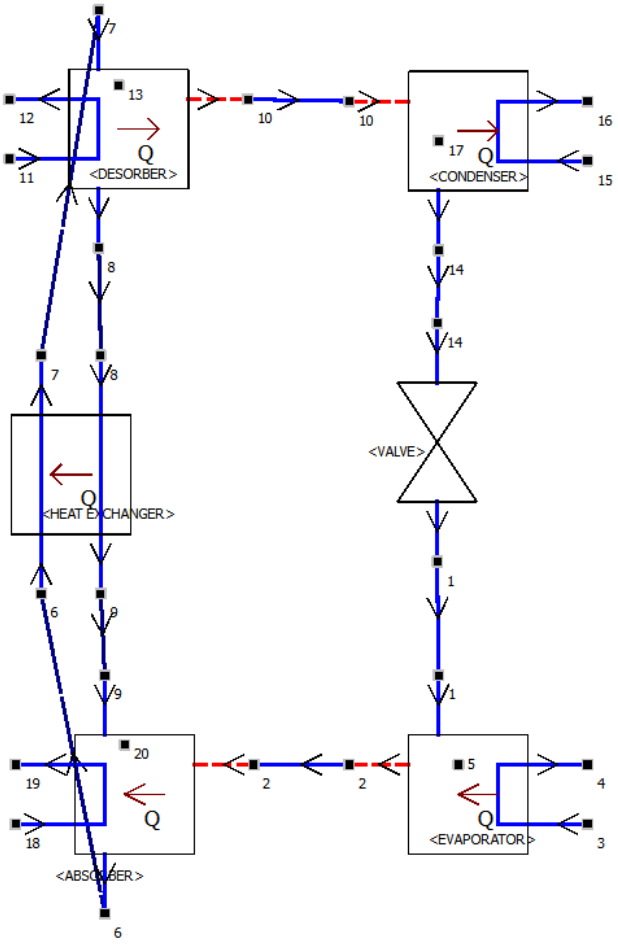

Fig B-1: SEC-Single-effect LiBr-water chiller

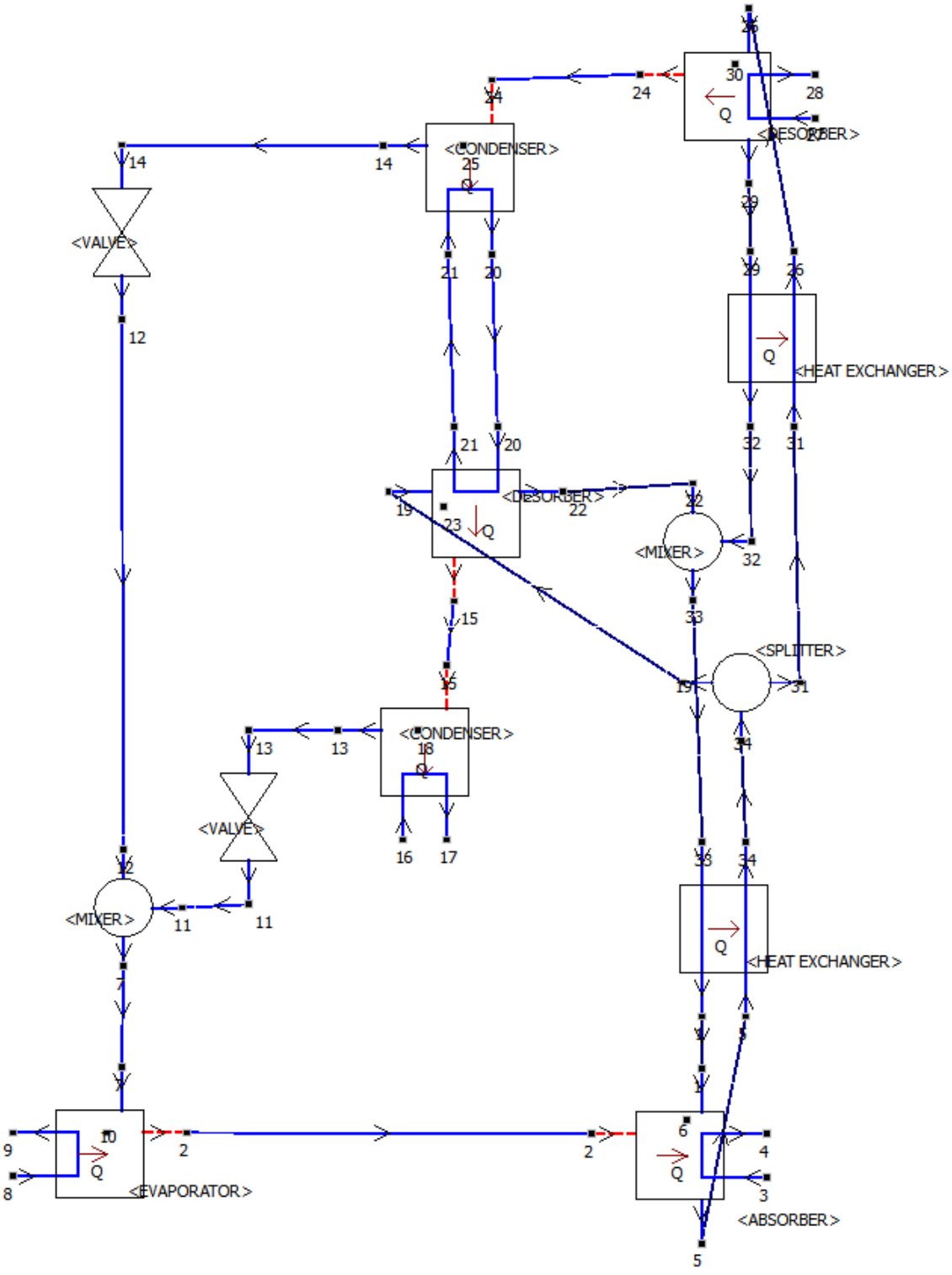

Fig B-2. DECP—Double-effect LiBr-water chiller, parallel flow

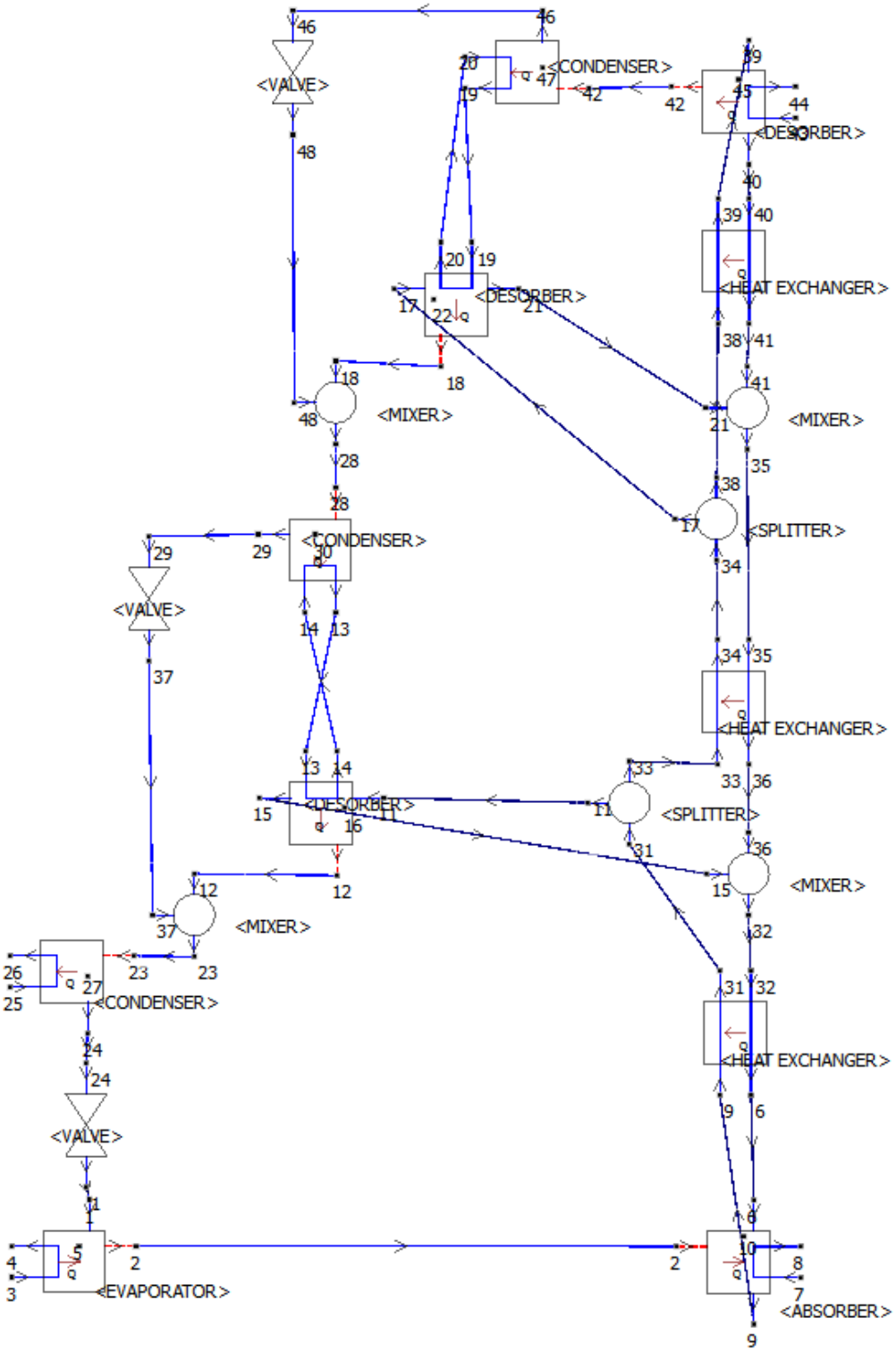

Fig. B-3: DAACP: Double-condenser-coupled, triple-effect chiller, parallel flow

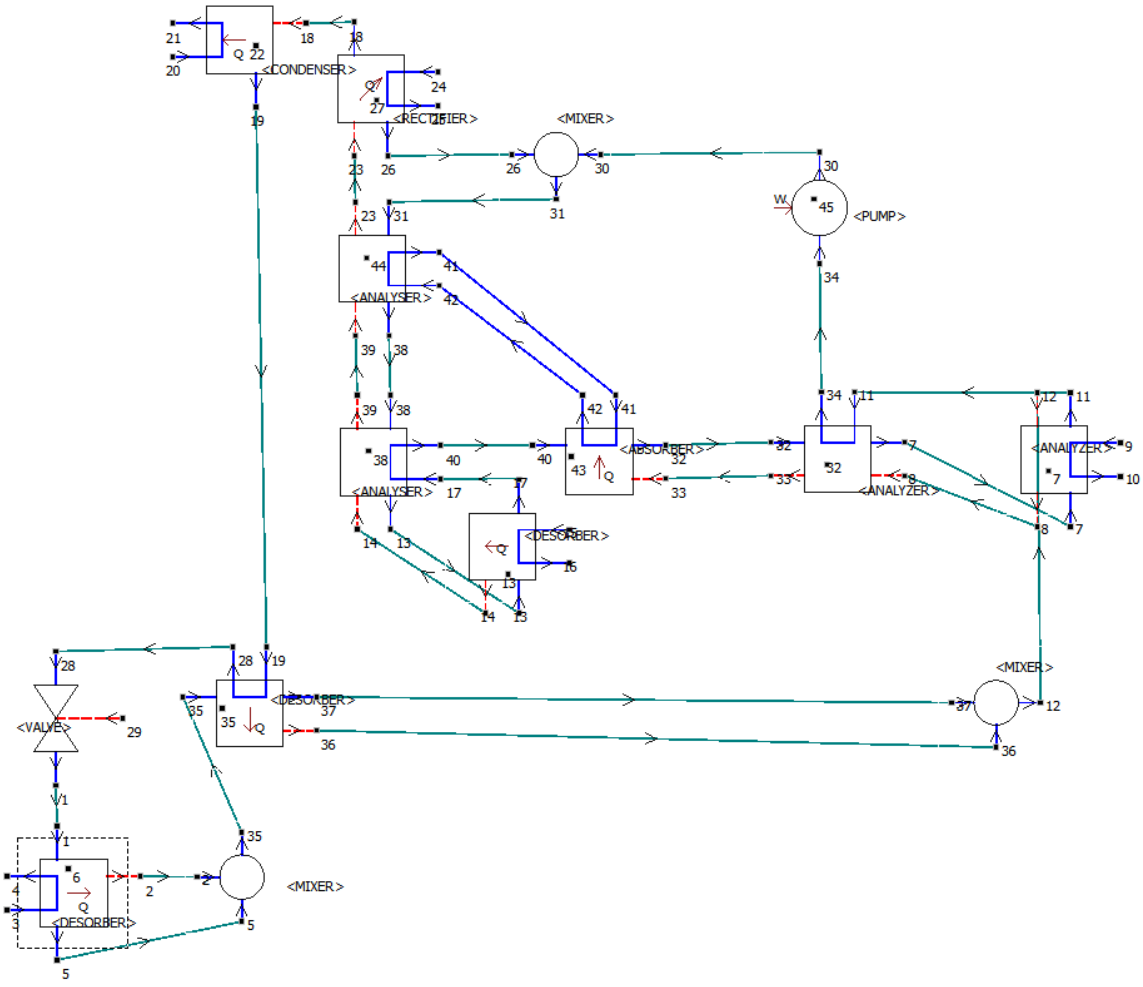

Fig. B-4: GAX—Generator-Absorber heat exchange, ammonia-water heat pump

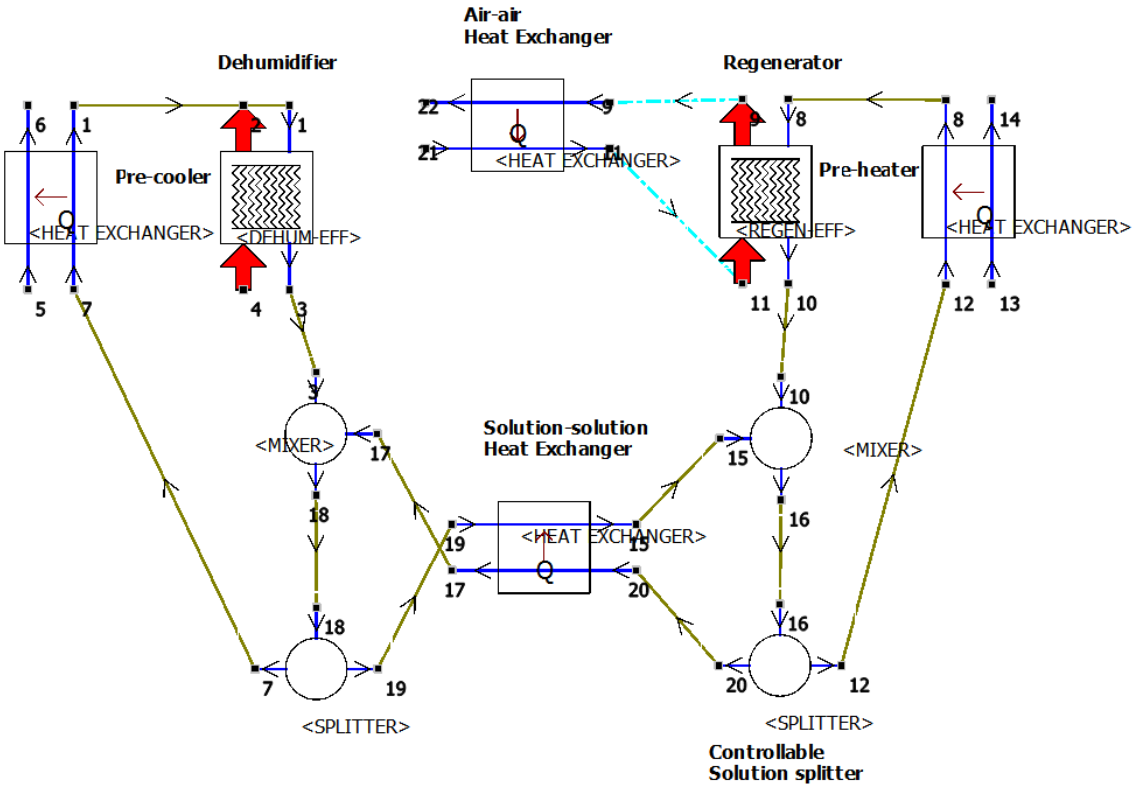

Fig. B-5: LDAC—Adiabatic liquid desiccant cycle

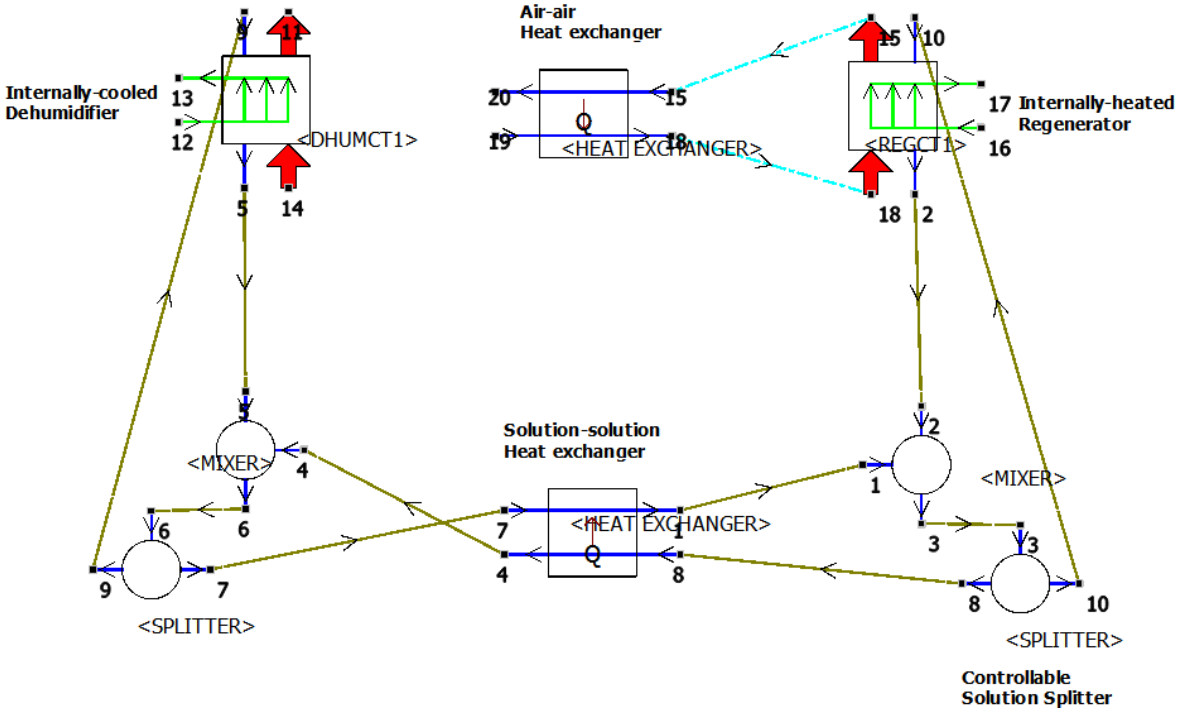

Fig. B-6: LDAC2—internally cooled/heated liquid desiccant cycle

REFERENCES

1. Ziegler, F., *State of the art in sorption heat pumping and cooling technologies.* International Journal of Refrigeration, 2002. **25**(4): p. 450-459.
2. Klein, S. and F. Alvarado, *Engineering equation solver.* F-Chart Software, Madison, WI, 2002.
3. PanahiZadeh, F. and N. Bozorgan, *The energy and exergy analysis of single effect absorption chiller.* International Journal of Advanced Design and Manufacturing Technology, 2011. **4**(4): p. 19-26.
4. Darwish, N., S. Al-Hashimi, and A. Al-Mansoori, *Performance analysis and evaluation of a commercial absorption–refrigeration water–ammonia (ARWA) system.* International Journal of refrigeration, 2008. **31**(7): p. 1214-1223.
5. Somers, C., et al., *Modeling water/lithium bromide absorption chillers in ASPEN Plus.* Applied Energy, 2011. **88**(11): p. 4197-4205.
6. Vidal, A., et al., *Analysis of a combined power and refrigeration cycle by the exergy method.* Energy, 2006. **31**(15): p. 3401-3414.
7. Grossman, G., *ABSIM-Modular Simulation of Absorption Systems, User's Guide and Refirenee*. 1995, Version.
8. Grossman, G., K. Gommed, and D. Gadoth, *A computer model for simulation of absorption systems in flexible and modular form.* ASHRAE Trans.;(United States), 1987. **93**(CONF-870620-).
9. Grossman, G. and M. Wilk, *Advanced modular simulation of absorption systems.* International journal of refrigeration, 1994. **17**(4): p. 231-244.
10. Grossman, G. and A. Zaltash, *ABSIM—modular simulation of advanced absorption systems.* International Journal of Refrigeration, 2001. **24**(6): p. 531-543.
11. Conde, M.R., *Properties of aqueous solutions of lithium and calcium chlorides: formulations for use in air conditioning equipment design.* International Journal of Thermal Sciences, 2004. **43**(4): p. 367-382.
12. Chaudhari, S. and K. Patil, *Thermodynamic properties of aqueous solutions of lithium chloride.* Physics and Chemistry of Liquids, 2002. **40**(3): p. 317-325.
13. Iyoki, S. and T. Uemura, *Heat capacity of the water-lithium bromide system and the water-lithium bromide-zinc bromide-lithium chloride system at high temperatures.* International journal of refrigeration, 1989. **12**(6): p. 323-326.
14. Handbook, A.F., *American society of heating, refrigerating and air-conditioning engineers.* Inc.: Atlanta, GA, USA, 2009.
15. Blanchette, J. and M. Summerfield, *C++ GUI programming with Qt 4*. 2006: Prentice Hall Professional.
16. License, G.L.G.P., *GNU Lesser General Public License.* 1991.
17. Grosse-Kunstleve, R.W., et al., *Automatic Fortran to C++ conversion with FABLE.* Source code for biology and medicine, 2012. **7**: p. 5.