

A COMPUTER MODEL FOR SIMULATION OF ABSORPTION SYSTEMS IN FLEXIBLE AND MODULAR FORM

G. Grossman, D.Sc. K. Gommed
ASHRAE Member

D. Gadath

ABSTRACT

A computer code has been developed for simulation of absorption systems in a flexible and modular form, which makes it possible to investigate various cycle configurations with different working fluids. The code is based on unit subroutines containing the governing equations for the system's components. Those are linked together by a main program according to the user's specifications to form the complete system. The equations are solved simultaneously, and fluid properties are taken from a property database.

The code is user-oriented and requires a relatively simple input containing the given operating conditions and the working fluid at each state point. The user conveys to the computer an image of his cycle by specifying the different subunits and their interconnection. Based on this information, the program calculates the temperature, flow rate, concentration, pressure, and 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.

The program has been used successfully to simulate a variety of single-stage, double-stage, and dual-loop heat pumps and heat transformers, with the working fluids LiBr-H₂O, H₂O-NH₃, LiBr/H₂O-NH₃, LiBr/ZnBr₂-CH₃OH and more.

INTRODUCTION

The need for computer simulation models of absorption systems has become apparent with the renewed interest in these systems during the past two decades. The advantage of the absorption cycle has been recognized for heat pumping applications where heat may be utilized as a source of power in preference over mechanical energy. Thus, absorption heat pumps utilizing waste or solar low-grade heat have emerged along-side gas and steam-fired systems for residential, commercial, and industrial applications. A variety of advanced cycles in multistage configurations have been proposed to fully utilize the thermodynamic potential of the heat source at hand. The development of working systems based on these cycles and the desire to investigate novel cycles and working fluids created the need for effective and reliable simulation.

Several computer simulation codes of absorption systems have been developed and described in the literature [1-6]. The results of some of them have been compared with experimental data and found to be in good agreement. The above codes were all system-specific, that is, they were written to simulate one particular system with a particular design, flow arrangement, and working material. Their structure did not allow easy modification to model other systems. In order to do that, major parts of the program, particularly the iterative sequence, would have to be rewritten.

The objective of the present work has been to develop a flexible computer simulation code capable of simulating absorption systems in varying cycle configurations and with different working fluids. The need for such a code grew out of repeated attempts under absorption research programs to evaluate new ideas for advanced cycles and working substances and compare them with existing ones. An earlier paper [7] described the modular approach taken to achieve this goal and the early results with three cycle configurations and two working fluids.

This paper describes an improved version of the code with a considerably extended user-oriented simulation capability and applicability. As will be shown, the code may be used not only for evaluating 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.

PROGRAM STRUCTURE

The objective of developing a flexible and user-oriented simulation code led to two basic requirements with respect to the program's structure. First, the program 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, the input requirements had to be kept simple and straightforward.

Recognizing that each absorption system consists of a number of standard components (e.g., absorber, evaporator, desorber, condenser, etc.), the following logic was developed for the program: Each basic component is simulated by a unit subroutine providing a mathematical expression of the physics of that component. Each 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, thermodynamic equilibrium. The unit subroutine, when activated, calls upon a property data base 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 them together in a form corresponding to the user's specification. 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, and all the equations have been established, a mathematical solver routine is employed to solve the set of nonlinear equations simultaneously.

Figure 1 is a schematic description of the program's architecture. The user input conveys to the

program an "image" of the cycle to be simulated: the number and types of units contained in it and their interconnections and size or transfer characteristics, where applicable. The input must also contain the values of the parameter set fixed by the user, e.g., temperatures, flow rates, etc., at specific state points. The main program interprets the user 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 data base as illustrated. All the equations and variables are normalized to be brought to the same order of magnitude. The solver package is then activated, and the values of the unknowns calculated to a user-specified accuracy. Recognizing that the set of non linear equations may have more than one mathematical solution, a set of constraints is applied to direct the solver toward the physically valid solution. The output contains the temperature, enthalpy, flow rate, concentration, pressure, and vapor fraction at each of the cycle's state points as well as the heat duty and transfer characteristics of each unit.

In developing the simulation code, the following tasks had to be accomplished: (1) develop the program logic; (2) develop a mathematical simulation of the system's units, describing their physical behavior; (3) obtain property data for the working fluids and develop a data structure; (4) adopt a method of solution and develop the solver package; (5) develop a user-friendly interface; (6) Implement the above in an effective computer code; (7) validate the model with experimental data.

Each of the program's parts mentioned above and illustrated in Figure 1 will be described in detail and elaborated upon in the following chapters.

INPUT AND OUTPUT

In developing the program logic as described in the previous chapter, the main program was assigned certain tasks (Figure 1) intended to alleviate a great deal of the burden from the user. 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. Furthermore, since 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 his cycle, which eliminates the need to worry about the order of the variables and parameters. This chapter describes the form of input and output to the code. The function of the main program will be described in the next chapter.

The Input Data

The user is first required to prepare a schematic diagram depicting the cycle to be simulated in terms of the basic nine units recognizable by the code (Figure 1). A detailed description of these units and their governing equations is given later. The diagram should display schematically the system's breakdown into the basic units and their interconnections. We shall follow the example in Figure 2 showing a single-stage, LiBr-H₂O absorption chiller consisting of an absorber, a desorber, an

evaporator, a condenser, a heat exchanger and a valve. The user is required to label each state point and each unit, in an arbitrary order. In Figure 2, numbers in a circle have been used for units, and numbers with no circle indicate state points. Then, the user prepares an input data file as described next. The example data file for the cycle in Figure 2 is given in Table 1.

The input consists of the following three parts:

a. General Preliminary Input - containing a user-supplied problem title (first card), scaling parameters for normalization (second card), and array sizes and convergence criteria (third card). Default options are available for all the above. The code can operate either in SI or I-P units, as indicated by a proper flag on the third card. On the fourth card, the user lists the total number of units and state points for the problem (6 and 20, respectively, for the present example).

b. Unit Input - Consisting of two cards for each of the units in the cycle. The first card shows the unit number, type, and heat and mass transfer characteristics. The second card lists its state point numbers in the schematic diagram in an order corresponding to that of the unit subroutine. Thus, for example, unit number 1 in Figure 2 is listed in the input as type 5, which the program recognizes as an evaporator; its state point numbers are 20,3,1,2,18 corresponding to 1,2,3,4,5 in subroutine EVAP contained in the code (Figure 4).

Heat transfer characteristics of a unit may be specified by one of four methods. The user can supply either the UA (overall heat transfer coefficient times area) or the effectiveness, or the CAT (closest approach temperature) or the LMTD (logarithmic mean temperature difference).

A flag (1 through 4) ahead of the applicable quantity on the first card indicates which of the four methods has been chosen for the particular unit. Thus, for the example in Table 1, unit No. 1 (evaporator) has been characterized by a UA of 662.1; unit No. 2 (heat exchanger) has been characterized by an effectiveness of 0.72; unit No. 4 (condenser) has been characterized by CAT of 2.5°. The next integer flag indicates the location of the temperature pinch: -1 when at the cold end, +1 when at the hot end. In the example of Table 1, the pinch for unit 1 is at the cold end, that is, between state points 2 and 18. Further details are given later. Mass transfer characteristics of a unit are specified by a temperature deviation from equilibrium at the outlet (last number on the first card). In the example of Table 1, the deviation from equilibrium at the solution outlet from unit No. 5 (absorber) is specified as -2.0°.

c. State Point Input - Consisting of one card for each of the state points in the cycle. Each card lists the state point number, a code for the working fluid in it, and five pairs of numbers for the temperature, flow rate, concentration, vapor pressure, and vapor fraction in that state point. Each pair of numbers consists of an integer index and a real value. The former can be either zero or non-zero, indicating a fixed or variable quantity, respectively. The latter gives a fixed value or an initial guess for the quantity in question. In the example of Table 1, state

point No. 1 (chilled water inlet to the evaporator) has a fluid code 3 (for water), a fixed temperature of 55.0, a fixed flow rate of 402.1, a fixed concentration of O.O, a fixed vapor pressure of O.O (as a subcooled liquid), and a fixed vapor fraction of O.O; State point no. 10 (vapor outlet from the desorber) has a fluid code 3 (for water), an unknown temperature with an initial guess of 60.0, an unknown flow rate with an initial guess of 3.7, a fixed concentration of O.O, an unknown vapor pressure with an initial guess of 3.0, and a fixed vapor fraction of 1.0. The same non-zero index greater than unity given to the same variable quantity at two different state points indicates that those two, although unknown, have the same value.

To summarize, by this form of input the user has conveyed to the program the complete necessary information about this 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.

The Output

The output structure is modular, aimed at giving the user control over the extent of the information presented. The integer flag MSGLVL in the third card of the General Preliminary Input allows the user to specify how frequently values of the variables and equation residuals at intermediate iterations should be printed.

Table 2 shows a sample output file for the example of Figure 2 and Table 1. This file was printed with MSGLVL = 0 and, hence, does not show any intermediate results. The initial section reproduces the input file as read by the program. This input-return enables the user to debug input errors by inspection. Next, the program prints the total number of iterations it took to converge and a convergence code (IER) indicating whether the required tolerance has been achieved. The next section contains the values of the temperature, enthalpy, flow rate, concentration, pressure, and vapor fraction at all state points. In the final section, the heat transfer characteristics and heat duty are given for each unit.

When MSGLVL > 0, additional information is printed. Table 3 describes a sample output file for the example of Figure 2 with MSGLVL = 100. Following the initial section with the input-return, pre-processed values of the temperatures at all state points are presented. These values have been reached by applying the constraints to the initial guesses for the temperatures to produce a feasible set-off point. Next is a printout of the total number of variables and equations and their respective types. The following section is again optional and consists of intermediate values of the physical quantities at all state points, along with the equation residuals. This information is presented in a table; the printing frequency is set by the value of the integer flag MSGLVL. Finally, the values of all dimensionless variables and equations residuals upon convergence are printed following the convergence code.

THE MAIN PROGRAM

The principal tasks performed by the main program are: (1) Interpreting the user input (2) forming a variable vector (3) linking the units and forming an equation matrix (4) Normalizing the variables (5) Activating the solver. The main program calls the unit subroutines and the solver package, when required, as illustrated in Figure 1.

By scanning through the State Point Input, the main program is able to identify the variable quantities by their non-zero flag. A variable quantity having the same non-zero flag greater than unity at two or more different state points is entered only once into the variable vector. The physical variables are normalized by division by the proper scaling parameters, taken from the second card in the General Preliminary 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 order 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 units interconnections. The equations are counted and each is assigned a number. During the solution process, the residues of the equations are calculated at successive iterations. The equations are normalized using the scaling parameters.

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 sometime 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.

UNIT SUBROUTINES AND GOVERNING EQUATIONS

Each of the system's standard units is modeled in a subroutine. The unit is treated as a black box, with its own inputs and outputs, that can be connected to other components. The governing equations take the unit as a whole and are formed from some or all of the following physical laws:

a. Conservation of mass for each material species (absorbent/absorbate) and/or conservation of total mass:

$$\sum_i (F_i C_i) = 0 \quad (1)$$

b. Energy balance:

$$\sum_i (F_i h_i) = 0 \quad (2)$$

c. Heat transfer, expressed in either of the following four forms:

$$Q_{UNIT} - UA * \overline{LMTD} = 0 \quad (3a)$$

$$\overline{EFF} - EFF = 0 \quad (3b)$$

$$\overline{CAT} - CAT = 0 \quad (3c)$$

$$\overline{LMTD} - LMTD = 0 \quad (3d)$$

where \overline{EFF} , \overline{CAT} , \overline{LMTD} are user-supplied values of the heat transfer effectiveness, closest approach temperature and logarithmic mean temperature difference, respectively, and EFF , CAT , $LMTD$ are calculated values of these quantities in terms of the temperatures at the unit's state points. When specifying a unit 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 \overline{CAT} and \overline{EFF} for each case. The definition of \overline{LMTD} is the same in both cases. By setting $IPINCH = 0$ the user lets the code decide the location of the pinch.

d. Vapor pressure - temperature-concentration (PTX) equilibrium between liquid and vapor:

$$f(P_i, T_i, C_i) = 0 \quad (4)$$

e. Mass transfer, expressed in terms of temperature deviation from equilibrium (DEV):

$$T_i = T_{iE}(P_i, C_i) + DEV \quad (5)$$

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

$$F_1(x_1, x_2, \dots, x_n) = \delta_1$$

$$F_2(x_1, x_2, \dots, x_n) = \delta_2$$

$$F_i(x_1, x_2, \dots, x_n) = \delta_i$$

(6)

$$F_n(x_1, x_2, \dots, x_n) = \delta_n$$

$$|\delta_i| \rightarrow 0 \text{ at the solution}$$

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 the physical solution regardless of the starting point. In order to do this, constraints were introduced into the program. The constraints are of the inequality type and are generated by the unit subroutines when called, in much the same way as the equations. Each unit has its own constraints, ensuring, for example, that the heat flow is from a hot to a cold stream and not vice versa.

The nine units recognizable by the code are described in Figure 4. These units were found sufficient to create most absorption cycles of interest.

MATERIAL PROPERTIES

The thermodynamic properties of working fluids have been incorporated in the code in the form of a property data base external to the unit subroutines. The purpose for 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 extension of the data base in the future and add properties of additional materials without affecting the rest of the program. Under this approach, each unit subroutine, when invoked, calls the data base 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 data base is illustrated in Figure 5. Two main subroutines, EQB and ENTHAL, form the interface between the property data base 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 . These, in turn, invoke further routines, which produce information on equilibrium temperatures, equilibrium vapor concentration liquid and vapor enthalpies, and the like for the material i . 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 the state points for that iteration.

The structure of each specific fluid routine varies according to the nature of the substance and data available. Originally, it was attempted to use a uniform data structure for all fluids, which turned out to be difficult and ineffective. Some materials may be easily described by a single polynomial-type formula in the entire range of interest for a particular property; others require a very complicated and cumbersome formula to cover only part of the range, and the roundoff error resulting from substitution in such a formula is prohibitively large. 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. It was therefore decided to use whatever form was most convenient and accurate for each material in each specific fluid routine.

The property data base presently contains the following materials:

material code -----	material -----
1	LiBr-H ₂ O solution
2	H ₂ O-NH ₃ solution and vapor
3	H ₂ O liquid and vapor
4	LiBr-H ₂ O-NH ₃ solution
5	LiBr/ZnBr ₂ -CH ₃ OH solution
6	CH ₃ OH liquid and vapor
7	LiNO ₃ /KNO ₃ /NaNO ₃ -H ₂ O solution

Following is a detailed description, by material, of the properties in the data base.

LiBr-H₂O solution (Code 1)

This solution has a nonvolatile absorbent, the salt LiBr, and uses H₂O (code 3) as an absorbate. Properties for this material for the practical range were compiled a few years ago by McNeely [9] who compared data from more than 15 sources and developed the pressure-temperature-concentration (P-T-x) and enthalpy diagrams. The properties are also available in the form of mathematical formulas and were taken from Ref. [10].

H₂O - NH₃ solution and vapor (code 2)

Since it was not possible to find a simple enough formula to describe ammonia-water properties, it was decided to incorporate the data in the code in tabular form. Two sources of tabular data, Refs. [11] and [12] were compared to each other and found to be in good agreement for most of the range. It was decided to use the tables by Jennings [11] which are more detailed, making linear interpolation results more accurate. These tables give the equilibrium temperature, NH₃ content in vapor, liquid and vapor enthalpies as functions of pressure and NH₃ concentration in the liquid. They cover the entire range of ammonia concentrations and a pressure range from 1.45 psia to 406.2 psia. The enthalpy reference point is taken at 32F (0°C), at which the enthalpy of both pure ammonia and pure water is considered to be zero. Thus, the tabulation is compatible with conventional steam tables. Since some pure ammonia tables use a datum of -40F, a correction must be made for cross reference with such a table.

H₂O liquid and vapor (code 3)

As a pure substance, functional relations for properties of water do not include concentration. Ref. [13] presents H₂O property data in the form of steam tables as well as in functional form. The latter was chosen to be included in the database.

LiBr-H₂O - NH₃ solution (code 4)

The ternary solution is a rather complex fluid involving one nonvolatile (LiBr) and one volatile (H₂O) absorbent. The solution's vapor is an ammonia-water mixture (Code 2) which, while in equilibrium with the ternary liquid is generally superheated in NH₃-H₂O terms. The properties in the data base are for a solution with 60/40% LiBr/H₂O ratio by weight, with a varying amount of NH₃ from 0 to 100% of the solution content. The properties of this solution were thoroughly measured and documented by Radermacher [14], and were fitted into mathematical equations [15] consistent with the H₂O-NH₃ and LiBr-H₂O data described earlier.

LiBr/ZnBr₂ - CH₃OH solution (code 5)

This solution employs a salt mixture as a non volatile absorbent for the methanol (Code 6) absorbate. The properties in the database are for a solution with an equal number of moles of LiBr and ZnBr₂. The properties of this solution were measured by Renz [16] and Uemura [17] and further extended and correlated by Biermann [18].

CH₃OH liquid and vapor (Code 6).

As a pure substance, functional relations for methanol do not include concentration. Properties in the subcooled, saturated and superheated range were compiled and calculated by Biermann [18] from data measurements by Gibbard and Creek [19].

LiNO₃/KNO₃/NaNO₃ - H₂O solution (Code 7).

This solution employs a mixture of three nitrate salts as a non volatile absorbent of water for high temperature lifts beyond the range accessible by LiBr-H₂O. The solution was studied by Davidson and Erickson [20] who found an optimum ratio of the nitrate salts at 53 w/o LiNO₃, 28w/o KNO₃ and 19 w/o NaNO₃. Their property measurements were fitted into mathematical formulae [21] consistent with the properties of pure H₂O (code 3).

THE SOLVER

We have described the process by which the physical behavior of the absorption system to be simulated is translated by the code into governing equations. The equations are expressed in the form of functions whose residuals at the solution are to cancel. Some of the equations are highly non linear, and the set, therefore, has more than one mathematical solution. Also, the code often generates redundant equations due to multiple mass balances from different units within a closed loop. In order to solve, it is necessary (1) to identify and eliminate redundant equations, (2) apply constraints to direct the solution-finding routine away from the physically unacceptable solutions, and (3) converge to a solution by reducing the residuals down to a specified tolerance.

Dealing with redundant equations could be circumvented by choosing a solution method capable of handling an overdetermined system. Such methods use an optimizer, instead of an equation solver, which minimizes an objective function including the residuals in some form (e.g. the sum of their squares). When a solution does not exist, the objective function does not become zero but rather represents a "best attempt" to satisfy all the equations. Furthermore, a constrained optimizer method could be employed which would select the physically acceptable solution from the multitude of mathematical solutions.

The main drawback of an optimizer for the present code is that it can converge and provide a physically erroneous result in the case of an underdetermined system, without being detected. This would happen, for example, if the user made a mistake in preparing the input and did not provide enough information. A solver, on the other hand, would not converge if it had more unknowns than equations.

The present version of the code employs the routine HYBRID1 developed by Powell [22,23], designed to solve an unconstrained fully determined system of nonlinear equations. The routine begins from an initial guess supplied for all the unknowns and searches for the solution by taking

successive steps of restricted size toward it. The search direction is determined at each step using the Levenberg-Marquardt algorithm which combines the Newton-Raphson and steepest-descent approaches. This makes it possible to avoid complications due to singular Jacobians. The routine returns the current values of the unknown variables after each step, which makes it possible to evaluate the residuals. Convergence is considered successful when the routine is able to satisfy either or both of the following conditions in less than the maximum number of iterations allowed by the user: (1) The Euclidian norm of the residuals is less than the user-supplied tolerance FTOL (2). The difference between two successive iterations is less than the user-supplied tolerance XTOL. Upon termination, the routine returns a convergence code indicating the status of the solution.

HYBRID1 operates in an unconstrained manner, and constraints directing it toward the physical solution have to be applied externally. As mentioned earlier, the constraints in this program are linear, inequality ones, concerning the temperature values at different state points in the simulated system. They are defined internally in each unit subroutine, as described before, to ensure a physically meaningful temperature distribution in the unit.

The constraints are applied in the code in two stages: once in a pre-processor of the input to modify the user-supplied initial guess for temperatures and provide the solver with an improved starting point; and again, after each step taken by the solver toward the solution. The intermediate values of the unknown variables returned by HYBRID1 are examined after each step in light of the constraints, and any non-complying temperature value is changed to a permissible one. The procedure interferes with the normal course the system would have taken in an unconstrained manner and forces it toward the physically meaningful solution.

The constraint enforcement is handled in the code by calling subroutine CONS. Any noncomplying state point temperature value is shifted by a certain measure, specified in the code. The shifting is done by adding the above measure to a value too small, subtracting it from a value too large, or executing a combination of both until the inequality constraint is satisfied. This approach, although somewhat heuristic, was found to be sound and robust.

The matter of redundant equations is handled by the subroutine REDUN. It should be noted that the system is never truly overdetermined when the problem is correctly specified; the set includes redundant, but not conflicting equations. The mass conservation equations from all units are separated from the other equations and placed at the end of the equation array. Being linear in nature makes it possible to replace any of the mass balance equations with a linear combination of itself with the others, without altering the solution or the nature of the Jacobian. Since the redundant equations in the system come from among the linear (mass conservation) ones, it is possible to properly eliminate some of the linear equations whose index exceeds the number of unknowns. This is done by adding each one of them to all the preceding linear equations and then discarding it, without loss of information.

Having dealt with the requirements stated at the outset, the solver is supplied with a fully determined, well-defined set of equations and a feasible starting value. Moreover, the solver is guided by the external constraint enforcement to a physical solution, as the iterations are confined to the feasible region.

RESULTS AND VALIDATION

Prior to using the program extensively for analysis of various advanced cycles, several initial runs were made for validation against experimental data taken of working absorption systems. Unfortunately, good documented data are quite scarce. The literature describes a number of absorption systems developed during the past decade, 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 lithium bromide-water heat transformer for upgrading waste heat [24]. Seventy-three experimental runs were documented where the machine had operated under steady-state, stable conditions for one to eight hours. The results of the validation have been described in detail in Ref. [7]. Very good agreement was obtained between the simulation and experimental results.

The program has been used to simulate a variety of cycles of interest with the working fluids presently in the data base. Some typical results are given in the Appendix. Among the cycles simulated are a single-stage heat transformer (Figure A.1) and a single-stage chiller (Figure A.2) with different fluids; single-stage chiller with a rectifier (Figure A.3) for H_2O-NH_3 ; a double-stage heat transformer (Figure A.4) with $LiBr-H_2O$; and double-effect chiller (Figure A.5) with $LiBr-H_2O$. Other systems not shown here, which involve various variations of the above, have been simulated successfully.

In preparing the input for a particular system, the user must supply those operating parameters that would be fixed in a real situation by the operator of a machine. These include the external flows and inlet temperatures, as well as rates of pumps forming part of the system. For example, in the single-stage chiller of Figure A.2, the input must include the flow rate and inlet temperature of chilled water to the evaporator, cooling water to the absorber and condenser, and hot water or steam to the desorber, as well as the solution pumping rate. Over- or underspecification would lead to nonconvergence. Our experience has shown that in most cases, many of these fixed parameters may be traded off against other variables, one for one, and convergence would still be obtained. In the above example, the user may choose to fix both inlet and outlet chilled water temperatures and let the computer calculate the flowrate.

The code was found to be quite robust with all the above systems. Run times are of the order of several seconds on a mainframe computer. Most systems with a working fluid involving a nonvolatile absorbent usually converge without difficulty; some problems are still encountered concerning fluids with a volatile absorbent such as H_2O-NH_3 , due to the complex behavior of

these materials at large concentrations of the volatile component in the vapor phase. Efforts are now underway to overcome this problem.

CONCLUSION

A computer code has been developed for simulation of absorption systems in a flexible and modular form. The code is capable of simulating systems in various cycle configurations and with different working fluids. A modular program structure has been developed to this end. The program has been used successfully to simulate several advanced cycles of interest and validated against experimental test results.

NOMENCLATURE

A	=	heat transfer area, ft ² or m ²
C	=	absorbent/absorbate concentration, %w/o
CAT	=	closest approached temperature, F or °C
DEV	=	extent of deviation from equilibrium state, F or °C
F	=	mass flow rate, lbs/min or kg/sec
EFF	=	heat transfer efficiency, dimensionless
h	=	specific enthalpy, Btu/lb or kJ/kg
IPINCH	=	integer index indicating location of temperature pinch
LMTD	=	logarithmic mean temperature difference, F or °C
MSGVLV	=	output message level code
P	=	pressure, psia or kPa
Q	=	heat transfer rate, Btu/min or kW
T	=	temperature, F or °C
U	=	overall heat transfer coefficient, Btu/(ft ² ·F·s) or kW/m ² ·°C
w	=	mass vapor fraction, dimensionless

Subscripts

i, j or numerical = state point index

L = liquid

V = vapor

E = equilibrium

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, 811-823, (1982).
2. J.C.V. Chinnappa and H.A. Martin, Dual Mode Operation by Solar Energy of an ammonia-water two-stage cycle: A comparative study. Solar Energy, 18, 337-342 (1976).
3. 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, Georgia, May 1979, K.W. Boer and B.H. Glenn, Eds. 1, 720-724, Pergamon Press. (1979).
4. G. Grossman and K. W. Childs, Computer Simulation of a Lithium Bromide-Water Absorption Heat Pump for Temperature Boosting. ASHRAE Transactions, 89, part 1b, 240-248 (1983).
5. M.O. McLinden and S.A. Klein, Steady State Modeling of Absorption Heat Pumps with a Comparison to Experiments. ASHRAE Transactions, 91, Part 2b, 1793-1807 (1985).
6. 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).
7. G. Grossman and E. Michelson, A Modular Computer Simulation of Absorption Systems. ASHRAE Transactions, 91, Part 2b, 1808-1827 (1985).

8. G. Grossman, Adiabatic Absorption and Desorption for Improvement of Temperature-Boosting Absorption Heat Pumps. ASHRAE Transactions, 88, part 2, 359-367 (1982).
9. L.A. McNeely, Thermodynamic Properties of Aqueous Solutions of Lithium Bromide. ASHRAE Transactions, 85, Part 1, 413-434 (1979).
10. American Society of Heating Refrigerating and Air Conditioning Engineers, Equilibrium Charts for Aqueous Lithium Bromide Solutions, ASHRAE Handbook of Fundamentals, pp. 16.81-16.82 (1977).
11. B.H.Jennings, The Thermodynamic Properties of Ammonia-Water Mixtures: A Reassessment in Tabular Form. ASHRAE Transactions, 87, Part 2, 419-433 (1981).
12. R.A. Macriss, B.E. Eaking, R.T. Ellington and J. Hueber, Physical and Thermodynamic Properties of Ammonia Water Mixtures. Research Bulletin No. 34, Institute of Gas Technology, Chicago, Illinois (1964).
13. American Society of Heating, Refrigerating and Air Conditioning Engineers, Thermodynamic Properties of Water, ASHRAE Handbook of Fundamentals, pp. 6.7 - 6.12 (1977).
14. R. Radermacher, Working Fluid Combinations for Absorption Heat Pumps, Doctoral Dissertation, the Technical University of Munich, 1981. Also, R. Radermacher and G. Alefeld., Lithium-Wasser-Lösungen als Absorber für Ammoniak oder Methylamin, Brennst-Wärme-Kraft, 34 (Nr. 1), 31(1982).
15. Phillips Eng. Co., St. Joseph, Mi. Private communication.
16. M. Renz, Thermodynamische Eigenschaften des Ternären Stoffsystems Methanol-Lithium Bromide-Zinc Bromide, Ki Klima Kalte Heizung, 9, 411 (1981).
17. T. Uemura and S. Hasaba, Studies on the Methanol-Lithium Bromide-Zinc Bromide Absorption Refrigerating Machine, Reito, 44, 720 (1969).
18. W.J. Biermann, Advanced Absorption Heat Pump Cycles, ORNL/Sub/81-17499/1, Oak Ridge National Laboratory, (1984).

19. H.F. Gibbard and J.L. Creek, Vapor Pressure Data of Methanol from 288.15 to 337.65 K, J. Chem. Eng. Data, **19**, 308 (1974).
20. W.F. Davidson and D.C. Erickson, New High Temperature Absorbent for Absorption Heat Pumps. ORNL/Sub/815-22013/1, Oak Ridge National Laboratory (1986). Also Proceedings, 8th Annual Industrial Energy Technology Conference, Houston, Texas, June 17-19, 1986, Vol. 1, pp. 350-358 (1986).
21. M.R. Ally, Oak Ridge National Laboratory, Private Communication.
22. M.J.D. Powell, "A Hybrid Method for Nonlinear Equations", Numerical Methods for Non-Linear Algebraic Equations, P. Rabinowitz ed., Gordon & Breach, London (1970), pp. 87 - 114.
23. M.J.D. Powell, "A Fortran Subroutine for Solving Systems of Nonlinear Algebraic Equations", Numerical Methods for Non-Linear Algebraic Equations, P. Rabinowitz ed., Gordon & Breach, London (1970), pp. 115-161.
24. W.R. Huntley, "Performance test results of an absorption heat pump using low temperature (60°C) industrial waste heat. Proceedings, the 18th Intersociety Energy Conservation Engineering Conference. (IECEC), Orlando, Florida, August 1983, **4**, 1921-1926. (1983). Also ORNL/TM-9072, Oak Ridge National Laboratory (1984).

ACKNOWLEDGEMENT

The authors are grateful to W.J. Biermann, R.C. DeVault, and H. Perez-Blanco for many useful suggestions. The project was sponsored by Oak Ridge National Laboratory for the U.S. Department of Energy under contract No. 19X-89673V.