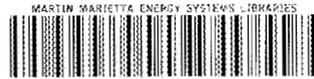


oml

**OAK RIDGE
NATIONAL
LABORATORY**

MARTIN MARIETTA



3 4456 0064192 8

ORNL/TM-9841

**Conceptual Design and Optimization
of a Versatile Absorption
Heat Transformer**

**H. Perez-Bianco
M. R. Patterson**

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
STACK ROOM 115
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this
report, send to name with report and
the library will arrange a loan.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes--Printed Copy: A04 Microfiche A01

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-9841

Energy Division

**CONCEPTUAL DESIGN AND OPTIMIZATION
OF A VERSATILE ABSORPTION
HEAT TRANSFORMER**

H. Perez-Blanco
Energy Division

M. R. Patterson
Computing and Telecommunications Division

Date Published-----June 1986

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under Contract DE-AC05-84OR21400



3 4456 0064192 8

CONTENTS

NOMENCLATURE	v
ACKNOWLEDGMENTS	vii
ABSTRACT	ix
1. INTRODUCTION	1
1.1 ABSORPTION CYCLES	1
1.2 SOFTWARE	6
2. CYCLE DESCRIPTION AND MODEL	11
3. COMPUTER SOLUTION	17
3.1 OPTIMIZATION CRITERIA	17
3.2 PROGRAM STRUCTURE	18
3.3 OPERATING MODES AND OUTPUT	21
4. RESULTS	25
5. CONCLUSIONS	39
REFERENCES	41
Appendix A: FIT OF ENTHALPY AND VAPOR PRESSURE FOR SODIUM HYDROXIDE-WATER SOLUTION	43
Appendix B: LISTING AND EXAMPLES OF OUTPUTS OF THE PROGRAMS	45

NOMENCLATURE

<i>C</i>	heat exchanger capital cost, \$/m ² (specific heat, kJ/kg·°C)
<i>DTLN</i>	arithmetic mean temperature difference, °C
δ	temperature difference, °C
<i>f_i</i>	function number <i>i</i>
<i>F</i>	fraction of the heat delivered in the lower absorber heat exchanger that is used as process heat
<i>h</i>	specific enthalpy, kJ/kg
<i>I</i>	installation cost factor
<i>m</i>	mass flow rate, kg/s
η	payback period, years
<i>O</i>	overhead factor
<i>p</i>	pressure, N/m ²
<i>Q</i>	heat flow, kW
<i>r</i>	heat of vaporization, kJ/kg
<i>S</i>	objective function
<i>T</i>	temperature, °C
<i>UA</i>	overall heat transfer coefficient times heat exchanger area, kW/K
<i>UAAO</i>	overall heat transfer coefficient times the absorber output heat exchange area, kW/K
<i>UAAR</i>	overall heat transfer coefficient times the absorber regenerative heat exchange area, kW/K
<i>UAC</i>	overall heat transfer coefficient times the condenser heat exchange area, kW/K
<i>UAE</i>	overall heat transfer coefficient times the evaporator input heat exchange area, kW/K
<i>UAGI</i>	overall heat transfer coefficient times the generator input heat exchange area, kW/K
<i>UAGR</i>	overall heat transfer coefficient times the regenerative generator heat exchange area, kW/K
<i>x</i>	sodium hydroxide weight per weight of solution, % (independent variable)
<i>y</i>	variable

Subscripts

<i>a</i>	absorber	<i>I</i>	inverse
<i>a,a</i>	absorber, adiabatic section	<i>o</i>	output section, overall
<i>c</i>	condenser	<i>p</i>	pressure
<i>e</i>	evaporator	<i>r</i>	regenerative
<i>g</i>	generator	1,2,3, ...	indications of state points or different functional dependence
<i>i</i>	input, dummy variable		
<i>j</i>	dummy variable		

ACKNOWLEDGMENTS

The support of the Department of Energy Office of Industrial Programs is gratefully acknowledged. The authors also wish to acknowledge the helpful comments and suggestions of G. Privon and R. Wail. J. W. Michel and S. I. Kaplan provided extremely useful input for the completion of this work. The help of Z. E. Whitson, L. D. Gilliam, and S. H. McConathy is warmly appreciated.

ABSTRACT

Heat transformers are absorption heat pumps that boost the temperature of industrial waste heat. Solutions of lithium bromide-water are commonly employed in heat transformers. Although these solutions have many desirable properties, they exhibit a drawback—a narrow solution field because of crystallization. The crystallization phenomenon limits the output temperature that can be obtained, particularly if a special type of heat transformer with high-temperature boosts is employed. This type of heat transformer has six heat exchangers instead of the customary five, and it has internal heat exchange between absorber and generator. Two questions then arise: is it possible to employ a working solution with a wider solution field than lithium bromide-water? If so, how can the heat transformer be designed? This report contains the theoretical results supporting the answers to those questions.

The choice of a solution of sodium hydroxide-water for this study was based on two facts: the solution does have a wide solution field, and its properties are well known. The six-heat-exchanger heat transformer is modeled in a digital computer, and this model is coupled to an optimizer. The optimizer allocates the heat exchanger size among the various heat pump components to produce a minimum payback period. The results show that when the waste heat and the heat rejection temperatures are low, sodium hydroxide-water shows operational advantages over lithium bromide-water. Otherwise, lithium bromide-water can be employed with basically the same results. The optimization results show relatively short payback periods (1 to 2 years), which indicate that the cycle is worthy of further study and experimentation. The design of absorption cycles via optimization techniques saves significant time and effort in specifying heat exchangers for a given set of desired operating conditions.

1. INTRODUCTION

1.1 ABSORPTION CYCLES

The interest in absorption heat pumps for industrial heat recovery is increasing, especially in those countries where economic conditions and the energy situation make energy efficiency a natural course to follow.¹ Reported on herein is an absorption heat pump, known as Type II or as a temperature booster or heat transformer, which has found applications in distillation columns. This heat pump inputs waste heat at a low temperature, rejects a fraction to a sink, and delivers the rest at a higher temperature for use as process heat. In the past, performance of prototypes has been modeled and tested,^{2,3,4} but there are no known commercial applications in the United States. As currently implemented, these heat pumps employ the conventional working fluids for large-tonnage absorption units, namely a solution of lithium bromide and water.

Although the lithium bromide-water solutions offer many desirable characteristics as working fluids, they also exhibit some drawbacks. Perhaps the most distinct drawback is the narrow solution field resulting from high crystallization temperatures. The crystallization phenomenon limits the maximum concentration at which temperature boosters are operated to about 65% in weight of salt per weight of solution.

This maximum concentration limit places an upper bound on the output temperature that can be obtained. Therefore, the study of other fluid combinations that do not exhibit this limitation seems appropriate. Of a number of proposed combinations, the solutions of sodium hydroxide and water are particularly promising.^{5,6} They exhibit lower crystallization limits than lithium bromide-water solutions, thus offering in principle the possibility of higher delivery temperatures.

A conventional lithium bromide-water temperature booster is illustrated in Fig. 1. The state points defined in this figure are indicated on a lithium bromide-water property diagram in Fig. 2. In this cycle, steam is generated in the evaporator (state point 9) by applying waste heat. The steam is absorbed in the absorber by a concentrated lithium bromide-water solution, releasing useful heat at a temperature higher than that of the waste heat (state points 1 and 3). This heat is transferred to an external stream employed for process heat. The diluted solution (state point 3) is cooled in a recuperative heat exchanger, and its pressure is reduced before it enters the generator (state points 5 and 6). In the generator, the solution is reconcentrated by evaporating water from it through the application of more waste heat. The solution leaves the generator (state point 6), its pressure is raised by a pump, and it returns to the absorber via the recuperative heat exchanger (state points 6 to 1). The steam leaving the generator is condensed (state point 8), and its pressure is raised by a pump before it enters the evaporator. Between points 1 and 1' and 5 and 5', a process of adiabatic absorption and desorption takes place. Although evidence shows that these processes occur very fast,⁴ a layer of packing as indicated in the design allows for residence time.

If higher delivery temperatures are desired, three options are available to the designer: to increase absorber pressure, to increase absorber concentration, or both. For instance, the absorber heat output of the single-effect cycle can be used to run a second

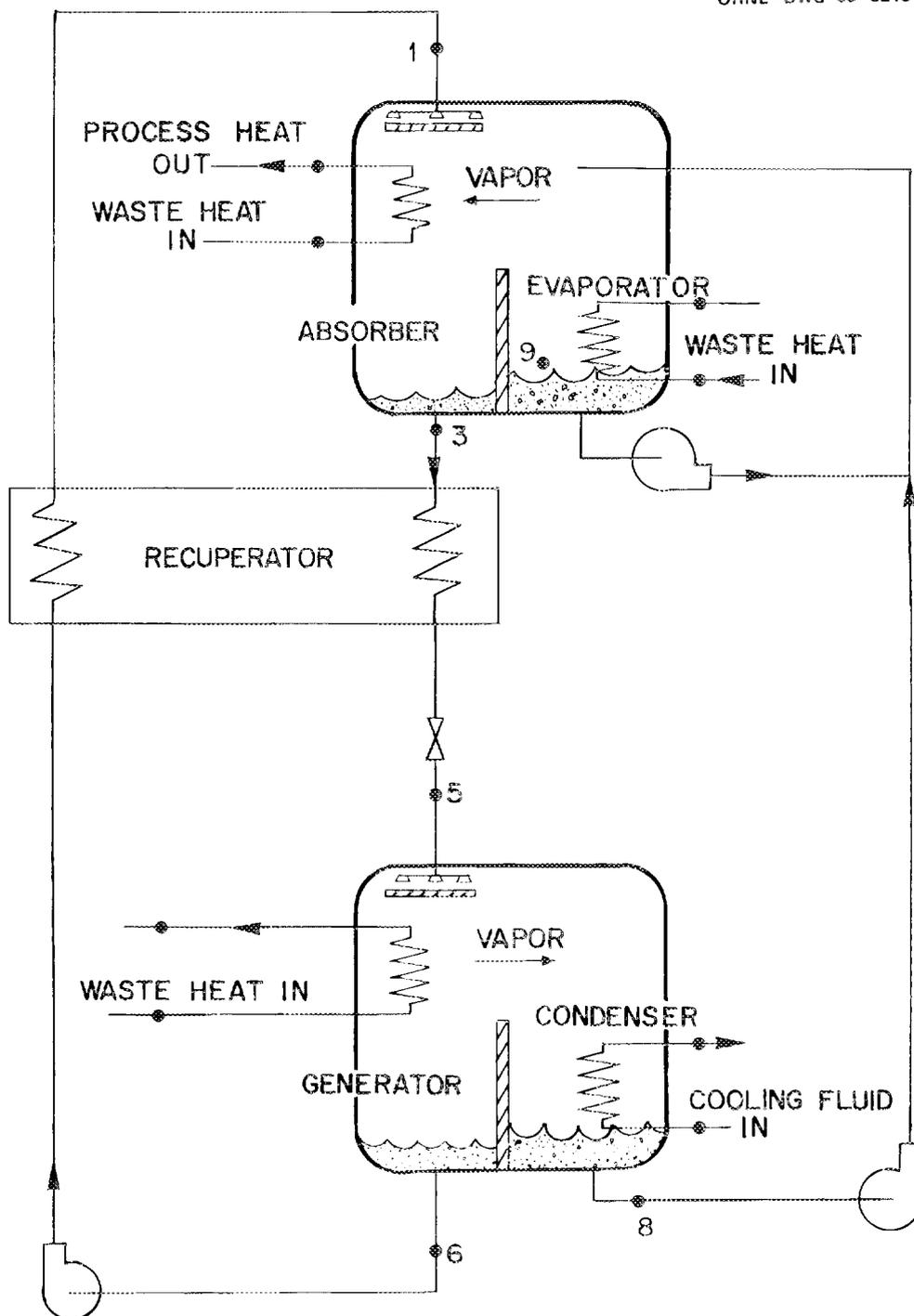


Fig. 1. Conventional lithium bromide-water cycle for temperature boosting of waste heat.

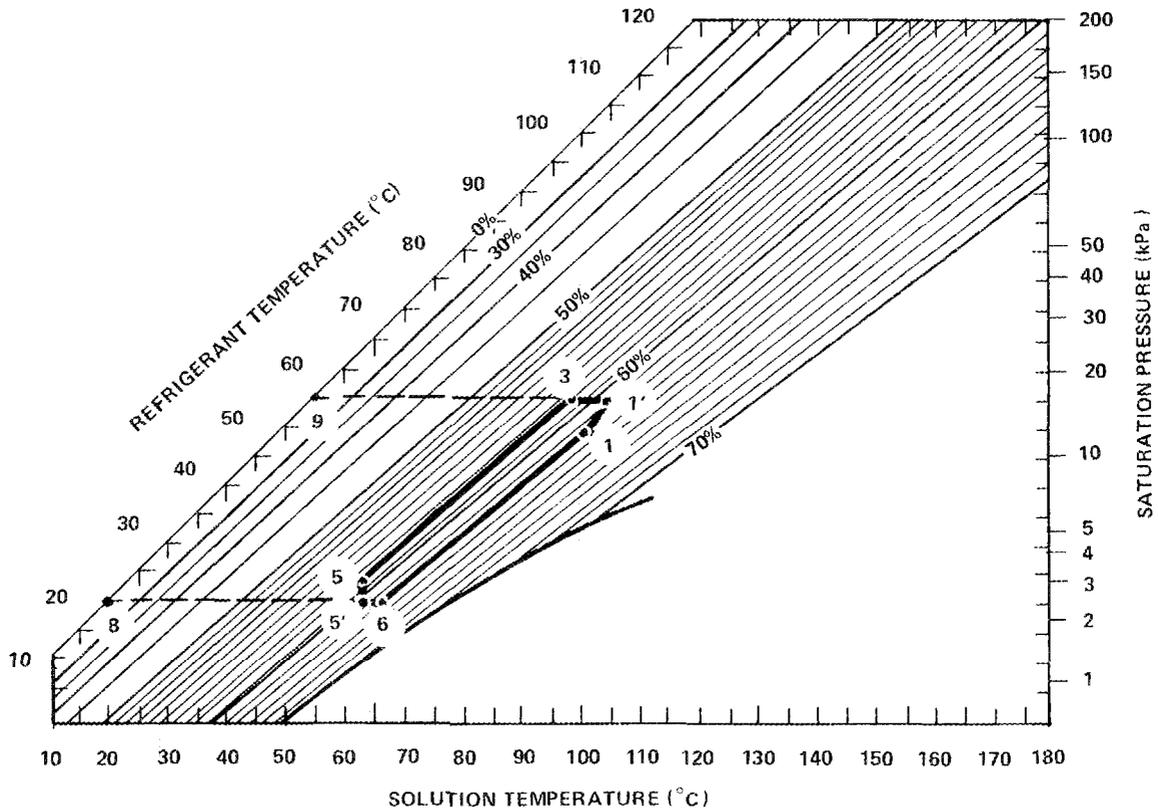


Fig. 2. State points for the cycle shown in Fig. 1. Percentages indicate concentration of lithium bromide by weight percent.

evaporator/absorber combination at the same maximum concentration as the first cycle. Shown conceptually in Fig. 3, cycles of this type have been extensively studied and tested with small variations.^{7,8} One should note that this option introduces a new pressure level in the machine. Consequently, a means of maintaining that pressure level, such as a new shell, must be provided. On the other hand, both absorbers are served by the same generator, which tends to reduce the capital costs associated with this option.

A second option is to run a completely different machine, with the output of the first one increasing both pressure levels and concentrations (Fig. 4). Whereas the new generator operates at the same pressure as the original one, the whole cycle operates at higher absorber pressures and concentrations. This forces the designer to provide ways of keeping the second solution loop separate from the first one, which in turn is likely to produce higher capital costs. Nevertheless, the maximum output temperature of this option is higher than that of the first option. The two options presented above involve the addition of new pressure and/or concentration levels. In addition, if it is desired to reach a temperature only slightly higher than that possible with a single-stage machine, the whole capital expenditure of a second stage must be considered.

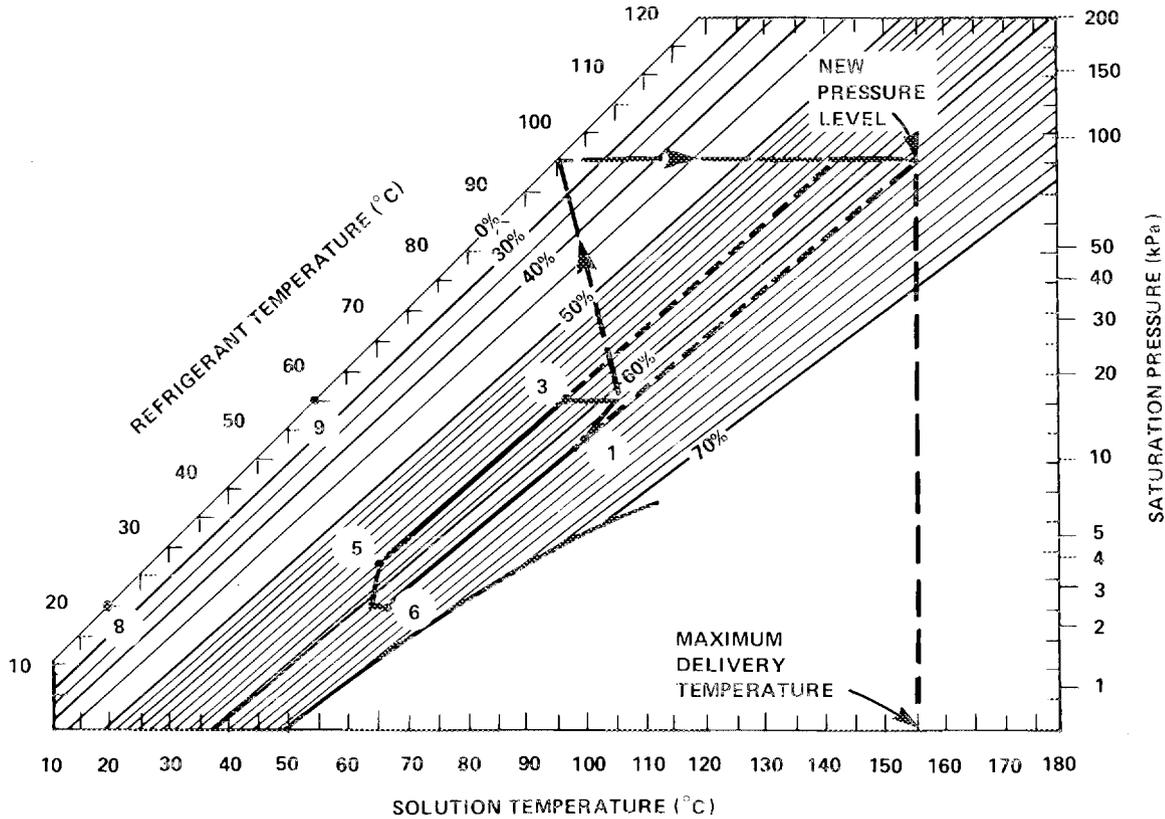


Fig. 3. State points for a two-stage cycle with increased absorber pressure. Percentages indicate concentration of lithium bromide by weight percent.

The third available option is simply to increase the cycle concentration at the same pressure levels as in the original cycle. In this option (Fig. 5), the absorber output is used to run a second-stage generator, but the absorbers of both stages are kept at the same pressure. The increase of the temperature boost is then only due to an increase in concentration. Although this option would at first sight appear as complicated as the two previously presented, the heat pump can be designed in such a way that relatively little additional hardware is needed. All that is required is some additional heat exchanger area in the cold area of the absorber; that heat is transferred to the hot area of the generator to effect the additional concentration. This is done by means of a regenerative loop. The machine then looks like the one shown in Fig. 6. To further increase the delivery temperature, more heat exchanger area may be added, progressively increasing the concentration. The concept does not call for additional pressure levels, and the needed heat exchanger area can be packaged in only two shells.

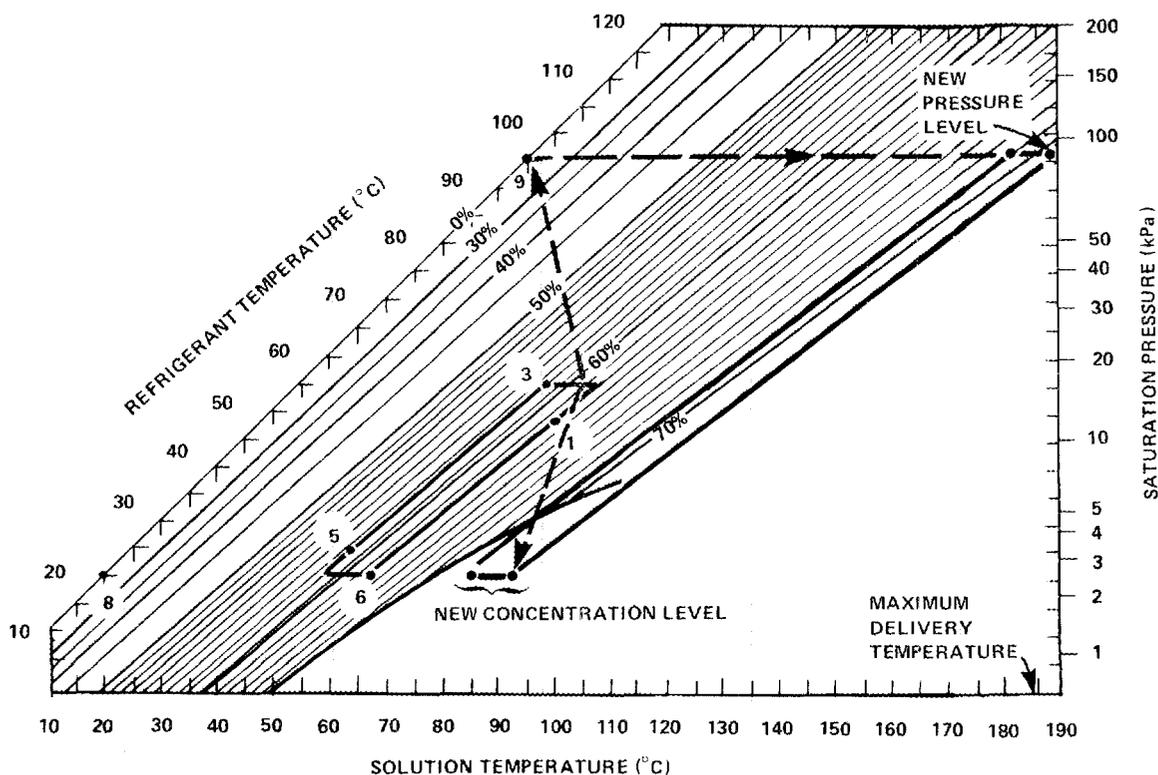


Fig. 4. State points for a two-stage cycle with increased pressure and concentration. Percentages indicate concentration of lithium bromide by weight percent.

The cycle shown in Fig. 6 must conform to the constraint $T_{24} \geq T_{20}$. This follows because as the solution trickles down the heat exchanger absorbing water, its equilibrium temperature decreases continuously. The case may then arise in which no waste stream at the temperature T_{24} is available. Also, heat at T_{24} (if it were available), could be employed in the evaporator to raise T_{25} even higher. In that case, a slightly different version of the cycle can be employed (Fig. 7). The waste heat input in the absorber is at the same temperature as in the generator, namely T_{22} . There are then two independent heat exchangers at the bottom of the absorber. One heat exchanger is part of the regenerative loop; the other simply preheats the fluid to be delivered as process heat from T_{22} to T_{24} . The three possible options are theoretical possibilities, and they constitute a subset of a large number of cycles that can be designed.⁹

The advantages of the concept of absorber-generator heat exchange (AGHX) for increasing the performance of residential heating and cooling absorption cycles have been thoroughly studied by B. A. Phillips.¹⁰ In application to heat transformers, the AGHX concept provides an increase in the possible delivery temperatures.

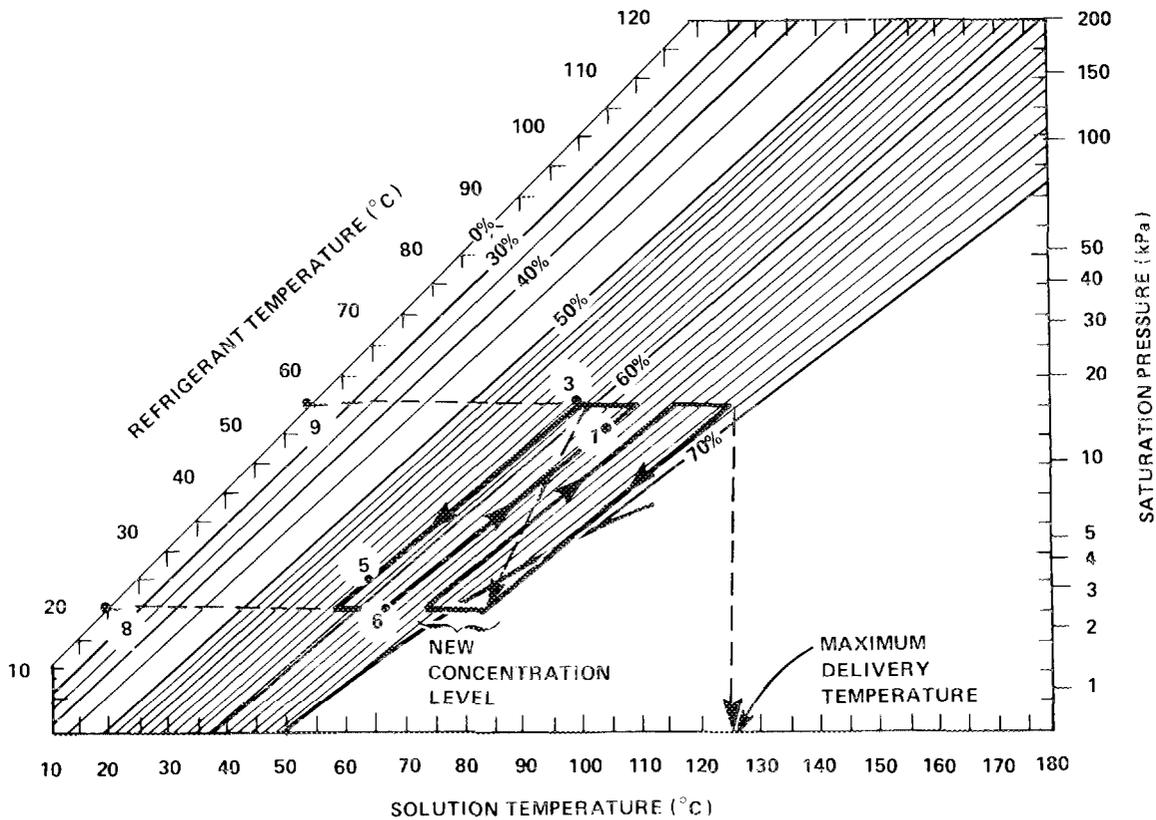


Fig. 5. State points for a two-stage cycle with increased concentration. Percentages indicate concentration of lithium bromide by weight percent.

In Figs. 4 and 5, one of the most important limitations of lithium bromide-water solutions is evident: the narrow solution field (a consequence of crystallization) prevents the development of the cycles described. Sodium hydroxide-water solutions exhibit a wider solution field than lithium bromide.⁵ The purpose of this report is to study the potential of sodium hydroxide-water coupled with the AGHX concept to increase the temperature of industrial waste heat streams. This fluid combination is not proposed here as the only alternative to lithium bromide-water solutions. However, because its properties are well known and it does have a wider solution field, study is appropriate to determine its full potential.

1.2 SOFTWARE

In this report, a mathematical model of the AGHX cycle is formulated. The resulting set of equations must be solved simultaneously, and numerical techniques requiring the use

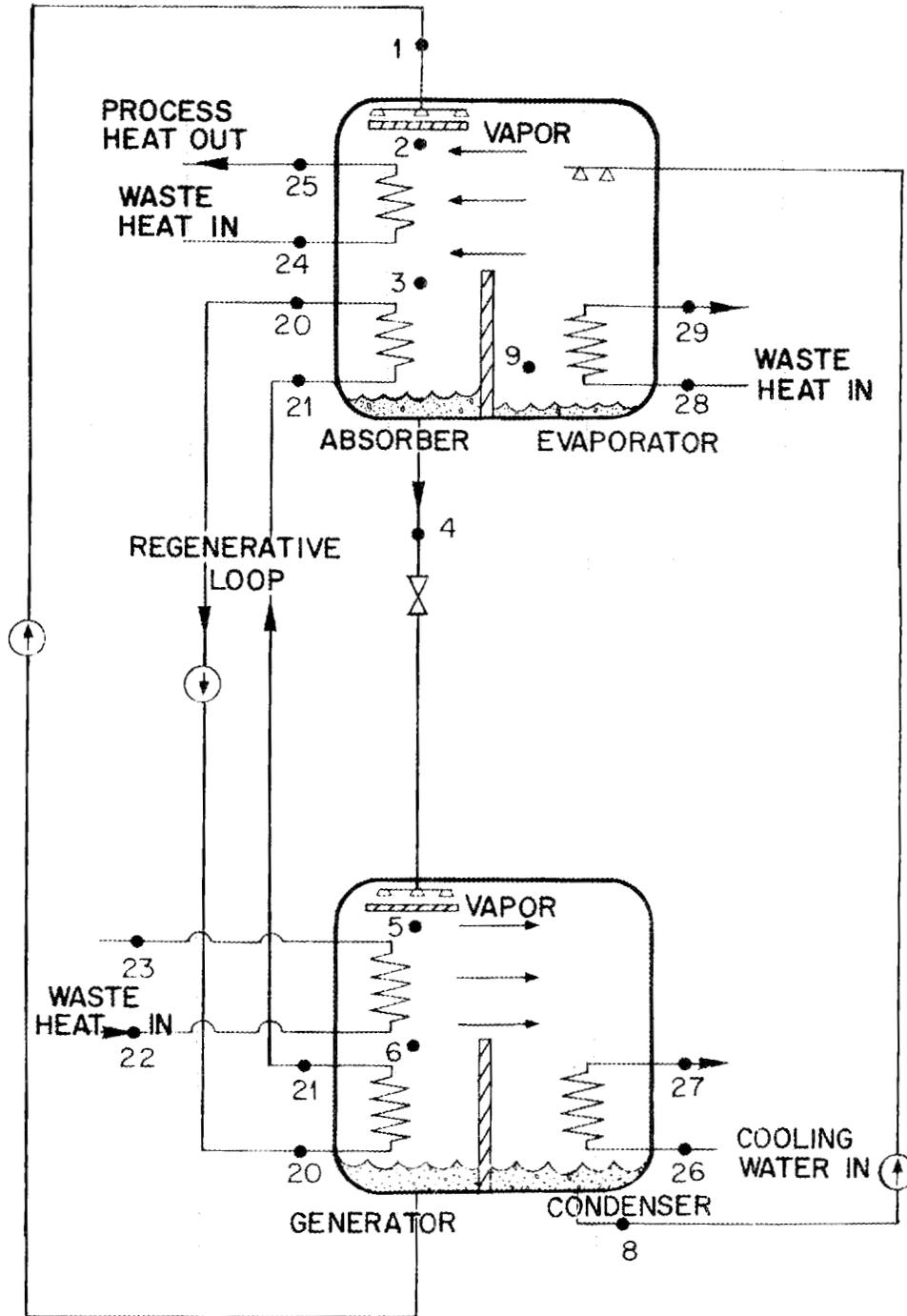


Fig. 6. Temperature boosting cycle with absorber generator heat exchange.

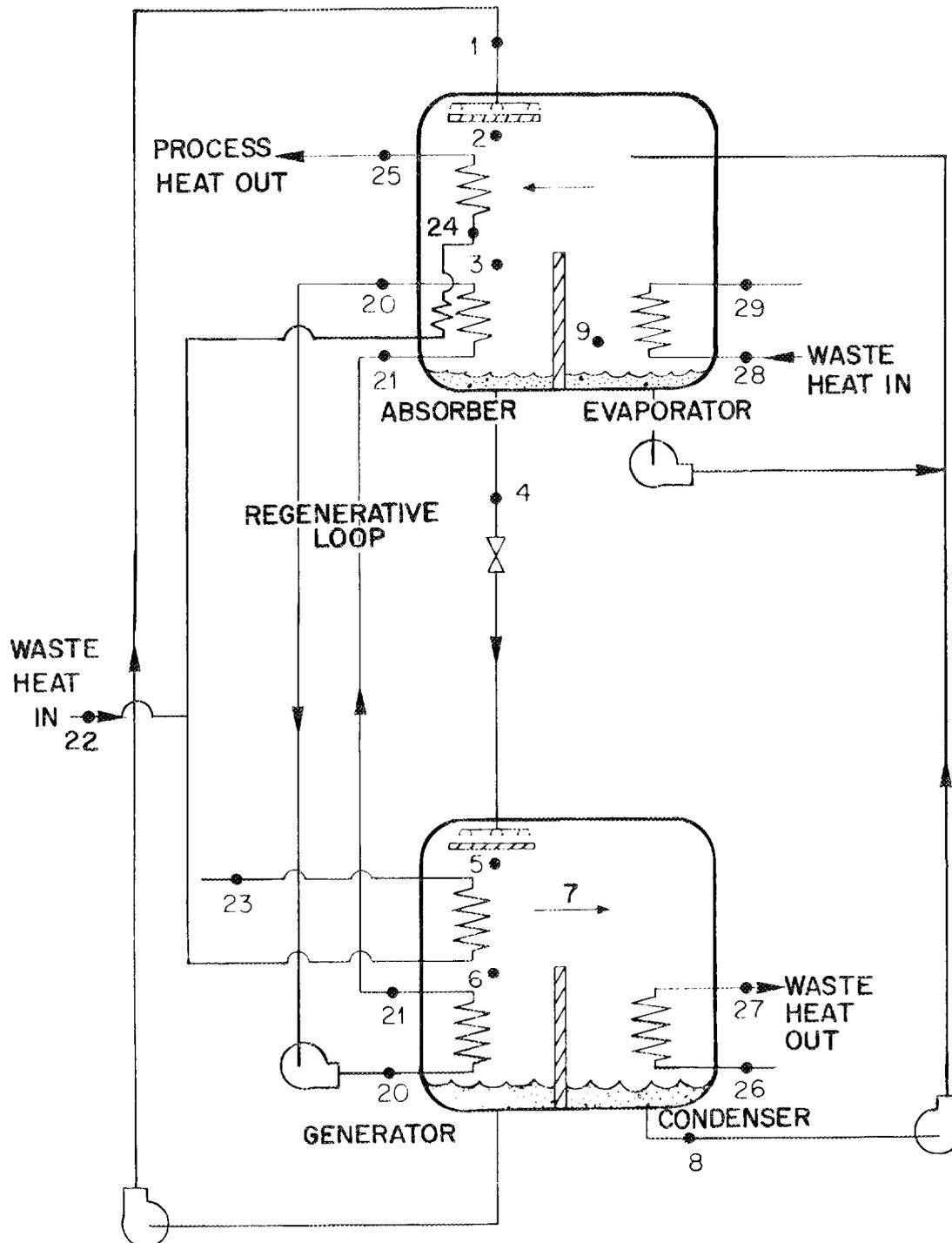


Fig. 7. Temperature boosting cycle with absorber generator heat exchange and two heat exchangers in the lower absorber part.

of a digital computer were employed. In addition, the computer model was coupled with an optimizer, which allows the user to minimize heat exchanger area or payback period.

This optimization program was written for the IBM 370/3033 in Fortran IV and uses 256K bytes of core and 140K bytes of virtual memory. No intermediate storage on disks or tape is required. The program executes in double precision because of the condition number of the matrix that is inverted during the solution. Its output is generally less than 5000 print records, including the printout of the source program but excluding the proprietary optimization routines. Because the running time is not large, less than 2 min, smaller computers could probably be used if they have the ability to carry double precision. The job is run as a batch job with no interactive programming required except in creating and editing the small amount of input data.

For the two different programs described in this report, two different matrix routines were used. The first, HYBRD1, is available on call from the CORLIB library¹¹ at ORNL and solves nonlinear equations. The second, E04UAF, is a proprietary routine and was furnished by the Numerical Algorithms Group (NAG).¹² It is part of a group of routines resident in the computer library at Oak Ridge National Laboratory. The names of these routines were chosen by NAG and apparently bear no relation to their function. The NAG routine used here is for optimization of an objective function with numerically determined partial derivatives and constraints on the variables. The E04UAF routine uses a sequential augmented Lagrangian method with a quasi-Newton method employed for minimization. The source subroutines for this part of the program cannot be included in this report because they were obtained under license from NAG. However, these routines are available from NAG for several machines. The transportability of the complete program is nevertheless limited by this consideration, and E04UAF should be replaced by a similar routine from the open literature as soon as one is available.

2. CYCLE DESCRIPTION AND MODEL

A schematic of a machine is shown in Figs. 6 and 7. The solution of sodium hydroxide absorbs water in the absorber from state points 1 to 4, delivering process heat from state points 2 to 3, and delivering heat to be recycled in the generator or for preheating from state points 3 to 4. The water vapor comes from the evaporator (state point 9), which is actuated by waste heat (state points 28 and 29). The solution pressure is reduced by a throttling valve, and the solution then enters the generator, where water is boiled off the solution by heat addition (state points 5 to 1). From state points 5 to 6, heat is added from waste heat (state points 22 and 23). From state points 6 to 1, heat recycled from the absorber is employed to further concentrate the solution. The water evaporated condenses in state point 8. The solution is pumped back to the absorber (state point 1), and the water is pumped to the evaporator (state point 9). From state points 2 to 3 and from 4 to 5, the solution undergoes a process of adiabatic absorption and desorption. The regenerative loop (state points 20 and 21) transfers heat from the cold part of the absorber to the hot part of the generator. This allows for a higher concentration at the generator exit, which in turn produces a higher delivery temperature.

To model this cycle mathematically, mass balances, heat balances, and heat transfer equations are used. Then, for the adiabatic section

$$m_2 = m_1 + m_{a,a} , \quad (1)$$

$$m_1 x_1 = m_2 x_2 , \quad (2)$$

$$m_1 h_1 = m_2 h_2 - m_{a,a} h_9 . \quad (3)$$

For the output section of the absorber (state points 2 and 3),

$$m_3 = m_2 + m_{a,o} , \quad (4)$$

$$x_3 m_3 = x_2 m_2 , \quad (5)$$

$$Q_{a,o} = m_2 h_2 - m_3 h_3 + m_{a,o} h_9 , \quad (6)$$

$$Q_{a,o} = UAAO \times DTLN (T_2, T_3, T_{25}, T_{24}) . \quad (7)$$

For the recycle absorber section (state points 3 and 4) and for the cycle shown in Fig. 6,

$$m_4 = m_3 + m_{a,r} , \quad (8)$$

$$x_4 m_4 = x_3 m_3 , \quad (9)$$

$$Q_{a,r} = m_3 h_3 - m_4 h_4 + m_{a,r} h_9 , \quad (10)$$

$$Q_{a,r} = UAAR \times DTLN (T_3, T_4, T_{20}, T_{21}) . \quad (11)$$

For the absorber recycle section (state points 3 and 4) and for the cycle shown in Fig. 7, the absorber section between state points 20 and 21 and between 22 and 24 is modeled as a single heat exchanger:

$$m_4 = m_3 + m_{a,r} , \quad (12)$$

$$m_4 x_4 = x_3 m_3 , \quad (13)$$

$$Q_{a,r} = m_3 h_3 - m_4 h_4 + m_{a,r} h_9 , \quad (14)$$

$$Q_{a,r} = F Q_{a,r} + (1 - F) Q_{a,r} , \quad (15)$$

$$Q_{a,r} = UAAR \times DTLN (T_3, T_4, T_{20}, T_{21}) . \quad (16)$$

Assuming constant specific heat of the fluid being heated in the absorber from T_{22} to T_{25} yields a relation between the heat recycled and the heat delivered as follows:

$$Q_{a,o} = C_p m (T_{25} - T_{24}) , \quad (17)$$

$$Q_{a,r} = C_p m (T_{24} - T_{22}) / F , \quad (18)$$

then

$$Q_{a,r} = \frac{Q_{a,o}}{F} \times \frac{(T_{24} - T_{22})}{(T_{25} - T_{24})} . \quad (19)$$

If steam is being produced instead of sensible heat, then it is assumed that the water is preheated from T_{22} to T_{24} (Fig. 7), and steam is obtained at T_{24} at the output of the absorber section. Then Eq. 20 is substituted for Eq. 19.

$$Q'_{a,r} = \frac{Q_{a,o} C_p (T_{24} - T_{22})}{rF} . \quad (20)$$

For the case shown in Fig. 6, the steam production is simulated simply by setting $T_{25} = T_{24}$. For the adiabatic generator section (state points 4 and 5),

$$m_5 = m_4 - m_{g,a} , \quad (21)$$

$$x_5 m_5 = m_4 x_4 , \quad (22)$$

$$m_4 h_4 = m_5 h_5 + m_{g,a} h_{g,a} . \quad (23)$$

For the input generator section (state points 5 and 6),

$$m_6 = m_5 - m_{g,i} , \quad (24)$$

$$x_6 m_6 = x_5 m_5 , \quad (25)$$

$$Q_{g,i} = m_6 h_6 - m_5 h_5 + m_{g,i} h_{g,i} , \quad (26)$$

$$Q_{g,i} = UAGI \times DTLN (T_{22}, T_{23}, T_6, T_{25}) . \quad (27)$$

For the recycle generator section (state points 6 and 1),

$$Q_{g,r} = m_{g,r} h_{g,r} + m_1 h_1 - m_6 h_6 , \quad (28)$$

$$Q_{g,r} = UAGR \times DTLN (T_{20}, T_{21}, T_1, T_6) . \quad (29)$$

For the condenser (state point 8),

$$m_8 = m_9 , \quad (30)$$

$$Q_c = -m_8 h_8 + m_{g,r} h_{g,r} + m_{g,a} h_{g,a} + m_{g,i} h_{g,i} , \quad (31)$$

$$Q_c = UAC \times DTLN (T_8, T_8, T_{26}, T_{27}) . \quad (32)$$

Note that in the above equation, the effect of the superheat of the vapor evolved from the solution is neglected. In the authors' experience, the resulting error is not significant.

For the evaporator (state point 9),

$$m_9 = 1 , \quad (33)$$

$$m_9 = m_{a,a} + m_{a,r} + m_{a,o} , \quad (34)$$

where m_a indicates absorber mass flow and the subindexes a , r , and o indicate adiabatic absorber, regenerative, and output sections, respectively,

$$m_9 = m_{g,r} + m_{g,i} + m_{g,a} , \quad (35)$$

$$Q_e = m_9 h_9 - m_8 h_8 , \quad (36)$$

$$Q_e = UAE \times DTLN (T_{29}, T_{28}, T_9, T_9) . \quad (37)$$

Equation 38 establishes the needed equality between the heat flows in the regenerating loop in the absorber and generator.

$$(1 - F) Q_{a,r} = Q_{g,r} . \quad (38)$$

Also, the following property relations were used for steam:

$$p_7 = f_5 (T_8) , \quad (39)$$

$$h_8 = f_6 (T_8) , \quad (40)$$

$$p_9 = f_5 (T_9) , \quad (41)$$

$$h_9 = f_4 (T_9) . \quad (42)$$

For the solution,

$$T_2 = f_2 (p_g, x_2) , \quad (43)$$

$$h_2 = f_1 (T_2, x_2) , \quad (44)$$

$$T_1 = f_{2,l} (p_7, x_1) , \quad (45)$$

$$h_1 = f_1 (T_1, x_1) , \quad (46)$$

$$T_3 = f_{2,l} (p_g, x_3) , \quad (47)$$

$$h_3 = f_1 (T_3, x_3) , \quad (48)$$

$$T_4 = f_{2,l} (p_g, x_4) , \quad (49)$$

$$h_4 = f_1 (T_4, x_4) , \quad (50)$$

$$T_5 = f_2 (p_7, x_5) , \quad (51)$$

$$h_5 = f_1 (T_5, x_5) , \quad (52)$$

$$h_{g,a} = f_3 [p_7, (T_4 + T_5)/2] , \quad (53)$$

$$T_6 = f_{2,l} (p_7, x_6) , \quad (54)$$

$$h_{g,i} = f_3 [p_7, (T_6 + T_5)/2] , \quad (55)$$

$$h_6 = f_1 (T_6, x_6) , \quad (56)$$

$$h_{g,r} = f_3 [p_7, (T_6 + T_1)/2] . \quad (57)$$

Finally, it was necessary to establish a relationship between T_{20} and T_3 and between T_{21} and T_6 . Without these relationships, the problem as posed above did not appear to always have a physically meaningful solution. Often it turned out that $T_{20} > T_{24}$ and $T_{21} > T_{20}$. Therefore, the following constraints were imposed on the model:

$$T_{20} \leq T_3 , \quad (58)$$

$$T_{21} \geq T_6 . \quad (59)$$

The above set of equations calls for property relationships for the steam and for the working solutions. These relationships for steam (f_5 , between temperature and pressure; f_4 , between temperature and enthalpy; and f_6 , between liquid temperature and enthalpy) are taken from ref. 13. The properties for sodium hydroxide-water (f_1 to f_3) were taken from ref. 14 and numerically fitted as explained in Appendix A. The corresponding property relations for lithium bromide-water were extracted from ref. 12.

3. COMPUTER SOLUTION

3.1 OPTIMIZATION CRITERIA

The current simulation code has two different forms. The first form of the code, using HYBRDI, solves the nonlinear equations that describe the process, yielding a set of operating conditions that correspond to the chosen values for input temperatures. Given a set of nonlinear equations

$$\begin{aligned} y_1 &= f_1(x,y) , \\ y_2 &= f_2(x,y) , \\ &\vdots \\ y_n &= f_n(x,y) , \end{aligned} \tag{60}$$

where $x = (x_1, x_2, \dots, x_N)$, the subprogram HYBRDI solves for the vector x that satisfies the system of Eq. 60.

The second form of the code approximates the solution of an augmented set of equations that has been expanded to include such quantities as the payback period. The minimization routine used in the code, a Numerical Algorithms Group (NAG) routine called E04UAF, allows one to constrain the variables such that, for example, temperatures are ordered in the way that one expects them to be: $T_2 > T_{25}$, etc. The solution is obtained by minimizing the sum of squares of the differences between left- and right-hand sides of equations that cannot be identically satisfied. The E04UAF routine minimizes the sum

$$S = \sum_{i=1}^N [y_i - f_i(x,y)]^2 + \eta_j^2 , \tag{61}$$

where the y vector is defined above and a single value of η was chosen to be the payback period.

$$\eta_1 = C \cdot I \cdot O / (4 \times 10^{-6} \cdot Q_{a,o} \cdot 3600 \cdot 8760) , \tag{62}$$

where

- C = capital cost of machine,
- I = installation cost factor (1.3),
- O = overhead factor (1.18),
- 3600 = seconds per hour,
- 8760 = hours per year,
- 4×10^{-6} = price of steam, \$/kJ,
- $Q_{a,o}$ = value given by Eq. 6 or Eq. 17.

Generally, the optimization form of the code works as well as the nonlinear equation solution, plus it allows other conditions to be superimposed, such as minimization of the payback. Because of this, the parameter space does not have to be searched manually to get the best set of operating conditions.

Selection of the operating conditions for a regenerative-loop heat pump is difficult for a new design such as the present one. Although a range of conditions may be expected for the key temperatures and pressures, exact values can be determined only by trial and error in the absence of a simulation model. This statement is especially true when the operating fluids are different from the conventional choices and have different thermodynamic properties. In this situation, a simulation model can provide insight into the settings for the control parameters and will give a reasonable approximation of operating conditions and performance. Simply by changing the thermodynamic properties, the code can be used for either sodium hydroxide-water or lithium bromide-water systems. Comparisons between these two types of systems can then be made on a similar basis, be it performance or economics.

3.2 PROGRAM STRUCTURE

The computer program consists of both locally written routines and proprietary routines. The main program, the routine NAOHEQ that sets up the equations to be solved, and the thermodynamic property routines for temperature and enthalpy were written by the authors. The routine E04UAF, along with its ancillary routines such as E04WAY, is a proprietary code provided by the Numerical Algorithms Group. Three routines, FUNCT1, CON1, and AMONIT were written locally to couple with the optimization routine.

The main code provides initially an estimate of the variables that are to be determined by the optimization code. In turn, the optimization code attempts to minimize the sum of squares shown in Eq. 61 for these variables. Approximately 14 variables are determined in this manner, depending on the configuration of the heat pump and the object of the run. A normal run takes from 2000 to 3000 iterations through the equations in NAOHEQ before it converges. During the run the routine CON1 maintains the constraints on the solution variables that are necessary to ensure proper relationships among temperatures and to keep the mass flows positive.

Although one has a choice of whether to use the solution by nonlinear equations HYBRD1 or by optimization E04UAF, the locally written codes are almost the same. The call to the solution routine in the main code is different, of course, but the remaining local routines are similar. Slight differences exist in NAOHEQ, but the property routines are unchanged. The routines FUNCT1, CON1, and AMONIT are not required for the nonlinear solution. A considerable advantage of the optimization code is that it does include the constraints set forth in CON1, which allow one to more nearly ensure that a realistic solution is obtained. The nonlinear equation solution is more direct and generally takes less computer time than the optimization code does. Overall, because of its flexibility and ability to constrain variables, the optimization method is favored.

Two separate programs, NAZERO and NAUOPT, call essentially the same set of routines to solve for the unknown variables in this problem (Fig. 8). The Fortran code for NAUOPT is included in Appendix B. In the case of NAZERO, a smaller set of variables is required because it determines only nine process quantities (m_{ar} , m_{ga} , m_{gi} , T_8 , T_9 , T_{25} , x_1 , m_1 , and $m_{a,o}$). The other code, NAUOPT, determines all of the above quantities except T_{25} , which is then fixed, along with the six heat transfer parameters $UAAO$, $UAAR$, $UAGR$, UAC , UAE , and $UAGI$. These composite UA values reflect the product of the overall heat transfer film coefficient and the heat transfer area.

The case of NAUOPT, which is similar to NAZERO, is described below. The process variables are either set within the program or they are determined internally in the code. Estimates of the nine process variables to be determined are set in the MAIN program. The variables are then prepared to be sent to the NAG optimization routine E04UAF by scaling them so that a point 80% of the way between the lower and upper limits is represented by unity to the optimization code. The parameters required by the optimization code are then set, including the number of constraint equations and the maximum number of iterations allowed, and the call to E04UAF is made.

Then E04UAF calls several locally written routines and other NAG routines. Principally, it calls the routine FUNCT1 to evaluate the scalar value of the objective function that it is designed to minimize. FUNCT1 in turn calls the routine NAOHEQ, which contains the equations that describe the heat exchanger. NAOHEQ takes the trial values that have been furnished to it via E04UAF from the main code and calculates the pressures, temperatures, mass flows, and concentrations that would exist in the remainder of the loop based on these conditions. With T_{25} fixed, eight equations cannot be satisfied in this manner, and the imbalance in each of these equations is used to form an element of the vector that is to be made zero. Each element is scaled so that its value is initially of order unity. These eight elements are returned to FUNCT1 where they are squared and summed to form the objective function. In this case, the objective function is augmented by the payback period, defined in Eq. 62. This payback period value is also scaled of the order of unity and is comparable with the scaled values of the other eight elements.

Control then returns to E04UAF, which calls CON1 to calculate the values of the constraint functions. The constraints serve the purpose of keeping all of the mass flows positive and ensuring that certain temperatures are larger than others. There are 14 constraint functions in the current program, and their values are printed only once at the beginning of the program for a checking purpose. Another routine called by E04UAF is AMONIT, the purpose of which is to monitor the progress of the solution by printing values of the condition number of the matrix intermittently during the solution. Various other NAG matrix routines such as E04WAY are called by E04UAF.

The subroutine NAOHEQ calls several routines so that various thermodynamic properties can be calculated. The following list gives these routines as they are called by NAOHEQ.

- F6—calculates the enthalpy of saturated liquid water at a given temperature.
- F4—calculates the enthalpy of saturated water vapor at a given temperature by first calculating the pressure at saturation and then calling the function F3 described below.

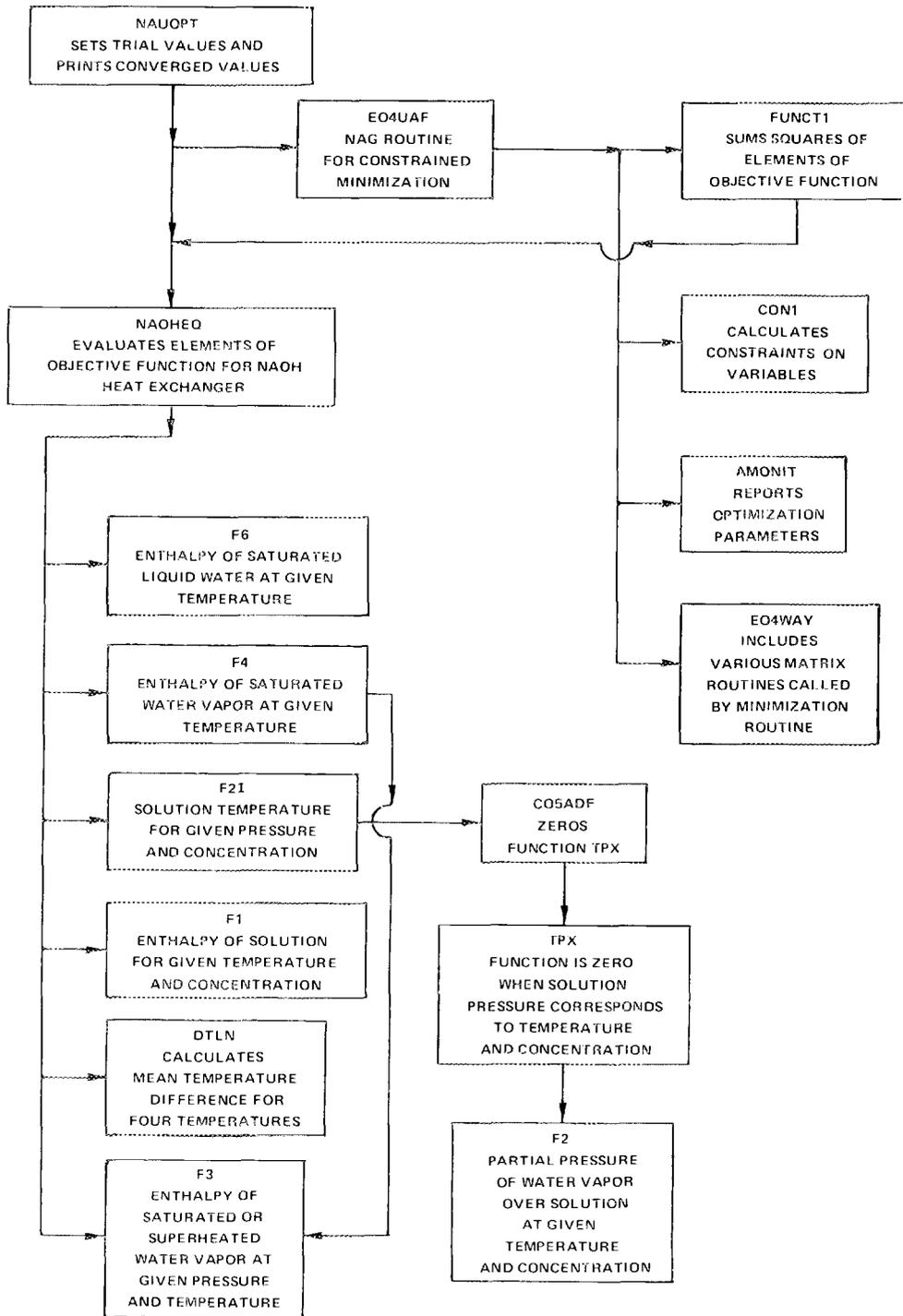


Fig. 8. Flow chart of the computer program.

F2I—calculates the solution temperature at a given pressure and concentration as the inverse of the process of calculating solution pressure from temperature and concentration. To do this calculation, F2I calls a library function C05ADF (ref. 12), which zeros a function TPX, which is zero only when the pressure corresponds to the temperature and concentration.

- F1—determines the enthalpy of the solution given the temperature and concentration.
- DTLN—calculates the mean temperature difference between one flow that enters at T_1 and leaves at T_2 and a second flow that enters at T_3 and leaves at T_4 . Conventionally, this mean difference would be calculated as the logarithmic mean temperature difference, but the present code uses the arithmetic mean temperature difference for purposes of simplicity. Difficulties with negative arguments to the natural logarithm (ALOG) function during early iterations of the solution led to adoption of the arithmetic mean. The arithmetic mean tends to make the temperature difference larger than the logarithmic mean, but the differences are not appreciably changed for the application in hand. Further, its use allows the solution algorithm to establish continuous functions and derivatives during the crucial initial portion of the problem.
- F3—calculates (from a direct call) the enthalpy of either saturated or superheated vapor at a given pressure and temperature.

On the first iteration and every hundred subsequent iterations, the routine NAOHEQ prints the values of the variables for the loop, including the value of the objective function. This information can be used to tell whether the optimization routine is properly directed toward a minimum. The information is also useful at the beginning of a run to determine whether all of the required data have been furnished and whether the initial guess for the variables to be determined is reasonable.

3.3 OPERATING MODES AND OUTPUT

The running time for the optimization code NAUOPT on the IBM 370/3033 for one case is about 1 min 10 s. In this run, approximately 2500 iterations are made in searching for the minimum value of the objective function. Occasionally a second run is necessary using the output of the first run as the input to the second run so that the estimates of the solution can be refined. This condition can be detected by markedly decreasing residual values of the objective function even after two or three thousand iterations in the first run.

The NAUOPT code is surprisingly short in length and consists of about 400 source lines of coding for the locally written part of the code. The length of the proprietary codes for E04UAF, the constrained optimization routines, is impossible for the authors to determine because the source coding is not available. This proprietary code was used because the open literature lacked codes that explicitly include the ability to handle constraints in a direct manner and do not require the user to furnish analytical gradients.

Excerpts from several runs are shown in Appendix B. Recall that NAZERO attempts a direct solution of the nonlinear equations by zeroing the trial vector. The program first prints $N=9$, indicating that nine equations are to be solved simultaneously, and then prints

the value of the return error, IER, which must be different from zero. Next, the input values for the guesses of the X vector are listed. Following that line are all of the variables given in the namelist \$NAM1. The number of iterations is given by the variable ITER at the end of this namelist. A second namelist, \$NAM2, follows and itemizes the elements of the X vector as transmitted to the subroutine NAOHEQ. These elements are again printed out with the statement **NAOHEQ WAS CALLED, X=**, followed by the elements of the vector X , then **Y=**, followed by the elements of the Y vector. These last values of the Y vector are the ones that the routine attempts now to zero.

The next iterations can be followed by observing the value of ITER. Ten iterations are required for the routine to make numerical estimates of the gradient, using step sizes that are too small to show in the number of decimal places printed here. In the eleventh iteration, the routine changes the elements of the X vector in an initial trial of a better solution. This process then continues until **ITER=83**, at which point the Y vector has been zeroed so that it satisfies the convergence criteria. The desired values, such as the coefficient of performance (COP), can be read from the same \$NAM1 namelist described for the first iteration.

An example of the output of NAUOPT is shown in Appendix B. Its organization is very similar to that of NAZERO, except that namelist \$SUBARG follows the first printout of the X vector. This set of variables is required by the optimization routine, and the list allows the user to verify that these are the quantities required by E04UAF. This routine has 28 arguments, many of which are arrays. The now familiar namelist \$NAM1 follows, and once again it includes most of the variables of interest for the cycle. The X vector is again listed with more precision in \$NAM2, and the message **NAOHEQ WAS CALLED, X=** precedes another listing of the X vector. Following **Y=** are the values of the elements of the scaled resultant vector, as for the previous program NAZERO. A value of the standard deviation of the Y vector elements is given following the label **SIGMA=**. The value of SIGMA is important because this program is set up to minimize this value.

The label **CON1: CC=** precedes three lines of output giving the 15 values of the inequality constraints for the first iteration only. These values are useful primarily for checking these relationships to ensure that they are being calculated correctly. The value of SIGMA will not decrease monotonically because each new trial X vector does not necessarily produce a better solution; however, each does allow a new part of the parameter space to be explored. Occasionally, the label **AMONIT: NITER, F, GLNORM, COND, RHO=** will appear, followed by five values on the next line. Only the fourth value is of any interest, **COND**, which is the condition number of the matrix that is being solved. Large values (10^{11}) are worse than smaller values (10^6); they can be observed but not controlled during the progress toward a solution. Larger values indicate that the optimization routine E04UAF may be unable to find a solution.

The unlabeled 15 quantities described above are repeated for each function evaluation that E04UAF calls. After each 100 iterations, the namelist \$NAM1 is printed so that the process variables and performance measurements can be viewed. The important thing is that the SIGMA value should generally decrease as the solution progresses. After the routine has either converged (**IFAIL=0**) or decided to quit (**IFAIL=2**) because it cannot do any better, namelist \$NAM1 is again printed along with namelist \$NAM2. These

values are also printed from the main program in \$XMAIN, along with repeat values for the X and Y vectors and the value of the objective function SIGMA. In normal use, the value of IFAIL will be 2 at the end because there usually is some refinement in the solution that the code could make given a much larger number of allowed iterations. Only occasionally will the code converge with IFAIL=0.

4. RESULTS

The set of results is classified into three subsets. The first one covers the potential of the cycle with sodium hydroxide–water solutions; the second one covers a comparison between solutions of sodium hydroxide–water and of lithium bromide–water; the third subset includes all the optimization results.

To learn about the potential of this cycle, NAZERO was employed. An arbitrary case was defined in which three streams of waste heat were assumed, one at 100°C returning at 60°C in the generator, another at 115°C returning at 107°C in the evaporator, and a third one at 140°C returning at 179.4°C in the absorber. All the chosen UA values and the temperature, concentration, and mass flow rate values corresponding to this case are shown in Fig. 9. The cycle COP, defined as the heat output in the absorber divided by the heat input in the evaporator and generator, namely

$$\text{COP} = \frac{Q_{a,o}}{Q_{g,i}}, \quad (62)$$

is equal to 0.31. This case shows that the cycle can accept a variety of waste heat streams and produce output temperatures on the order of 180°C. The output temperature T_{25} depends on the UAs . By increasing the $UAAO$ from 200 to 500 and $UAGI$ from 55 to 65, T_{25} is raised from 179.4 to 186°C. The possibility of feeding the evaporator with the 140°C stream was not studied.

Perhaps one of the most distinctive characteristics of this temperature booster is its versatility. Different waste heat temperatures can be accepted to produce a wide range of output temperatures. This can be done by varying the UA values of the regenerative loop. To gain insight into this capability, the UA values of the regenerative loop $UAAR$ and $UAGR$ were varied, keeping all the other temperatures constant; the values are shown in Fig. 10. Waste heat was assumed available at 90°C, and its temperature was dropped to 85°C in the evaporator and to 60°C in the generator. This relatively large temperature drop may make this cycle very attractive, since a common design problem of these machines is the limitation in the temperature drop necessary to keep concentration levels high. The results of this analysis are shown in Fig. 11, where the outlet absorber temperature T_{25} and the COP are shown vs the maximum cycle concentration x_1 . The values of $UAAR$ and the $UAGR$ are shown for each point. It is clear that output temperatures from 100 to 170°C are possible by increasing the area or the heat transfer coefficient (or both) in the regenerative loop. The COP decreases with increasing temperatures from 0.48 to about 0.41. The Carnot COP is also shown. As shown in previous research,¹⁰ the COP of the normal cycle tends to follow the Carnot COP. The reversed cycle studied here appears to be no exception. Apparently, operation of this cycle at high concentrations (greater than 70%) may call for a dilution cycle before shutdown to avoid crystallization of the sodium hydroxide solution. The values calculated for $UAAR$ of 280 kW/K and $UAGR$ of 310 kW/K are shown in Fig. 12. The corresponding states are shown in Fig. 13.

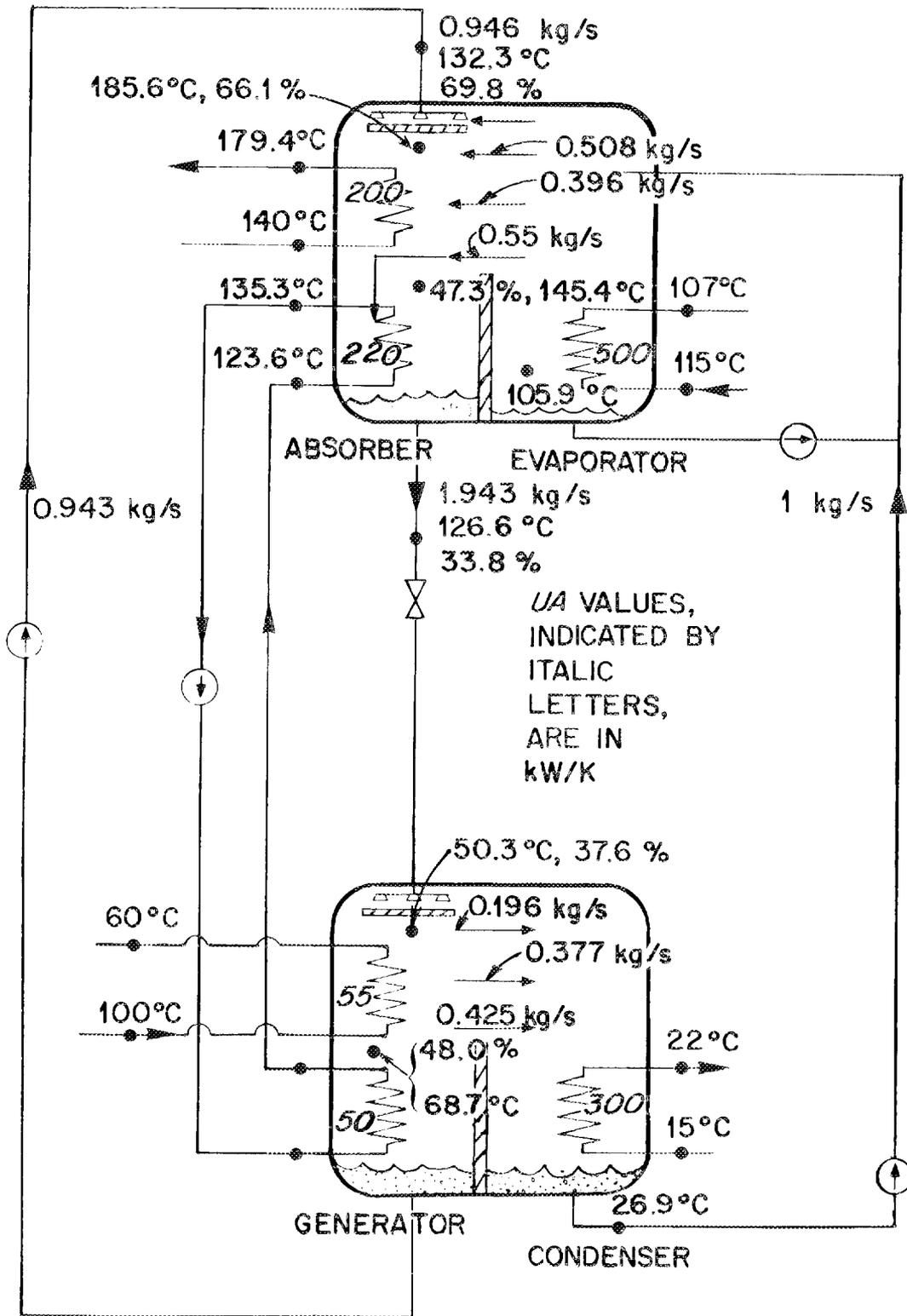


Fig. 9. Flow rates, temperatures, and concentrations for the indicated *UA* values; a delivery temperature of 180 °C may be reached.

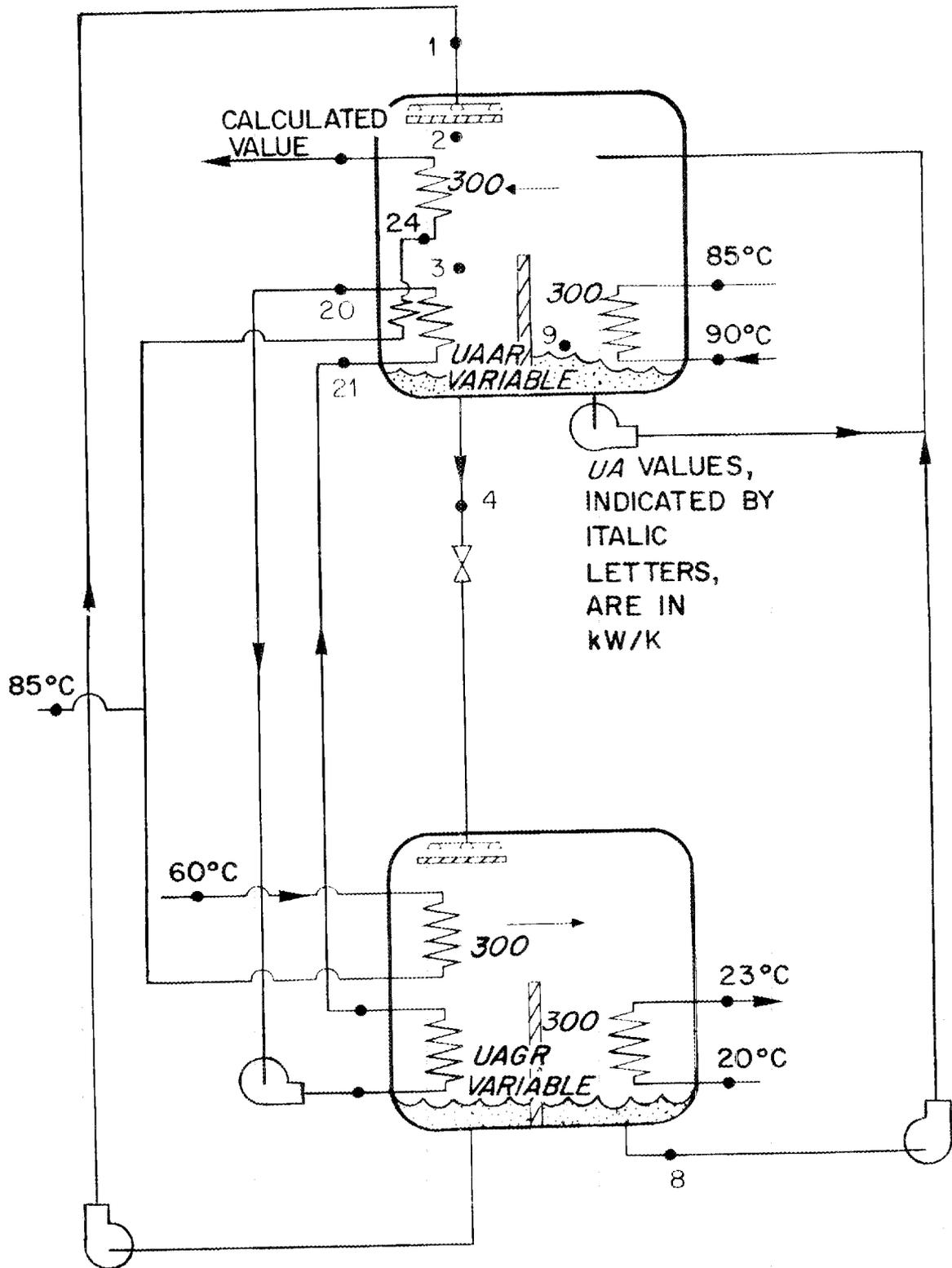


Fig. 10. Conditions for the runs shown in Fig. 11.

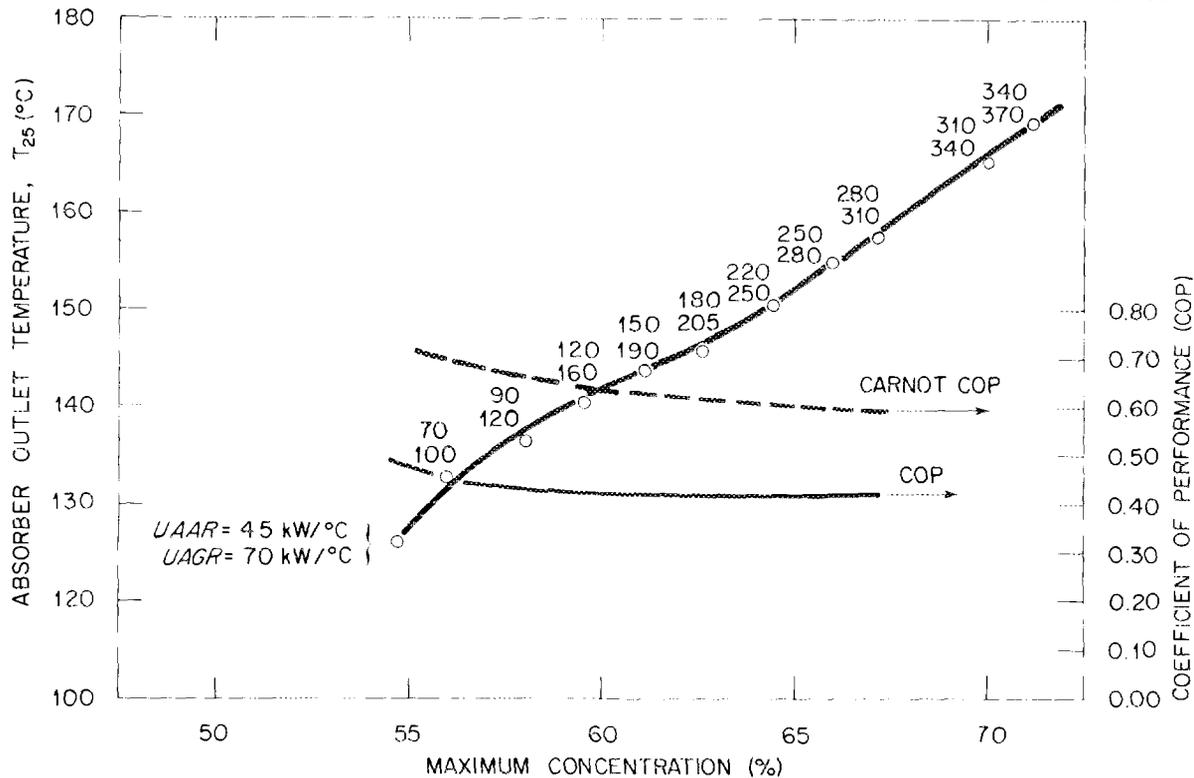


Fig. 11. Potential of the cycle shown in Fig. 7: delivery temperatures and COP (other data shown in Fig. 10).

To gain insight into the relative usefulness of this cycle implemented with lithium bromide as opposed to sodium hydroxide, three comparative cases were run for both absorbents. The UA values required to yield a given value of T_{25} with 100°C waste heat were determined by the optimizer, which minimizes payback periods. The results are shown in Table 1.

The payback periods (calculated for a heat exchanger price of $\$240/\text{m}^2$) may not be comparable because the construction materials will in all likelihood be different. However, the sum of the UAs and the COP values are an indication of the cycle performance with each fluid. The data corresponding to Case 1 show that the lithium bromide-water cycle requires either higher overall heat transfer coefficients or greater total heat exchanger area than sodium hydroxide to reach the same delivery temperature. The COP value is higher for lithium bromide. However, an inspection of the exit generator solution concentration and temperature values (x_1 , T_1) shows that the lithium bromide-water solution is on the verge of crystallizing. Any further increase in the delivery temperature T_{25} while the condenser temperature T_8 stays constant requires concentration values that are well within the crystallization region for lithium bromide.

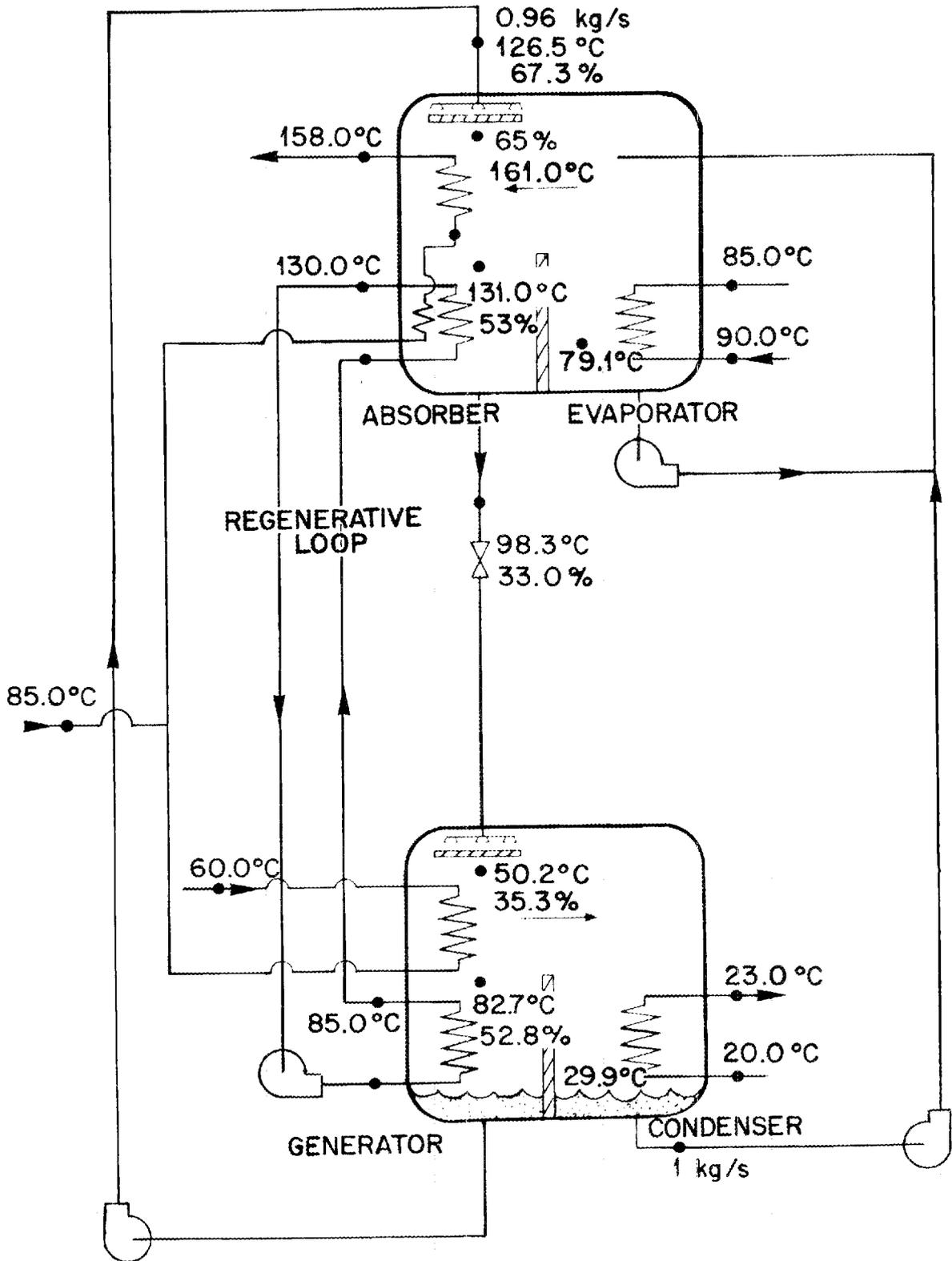


Fig. 12. Example from Fig. 11; case for $UAAR$ and $UAGR$ equal to 280 and 310 kW/K, respectively.

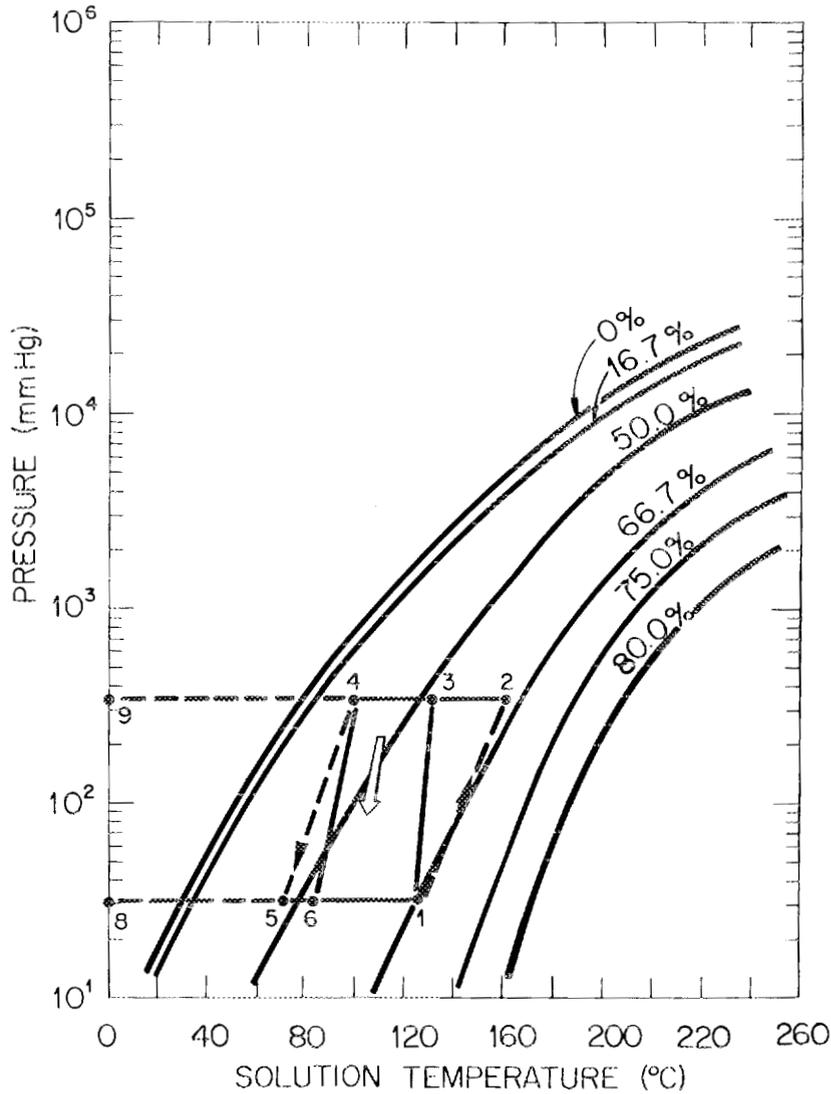


Fig. 13. State points corresponding to the cycle shown in Fig. 12 (data from ref. 12).

The effect of the crystallization-imposed limits are shown in Table 1, Case 2. Whereas a temperature boost of 55°C can be obtained with sodium hydroxide, the lithium bromide-water solution cannot operate because of crystallization. Raising the condenser temperature of the lithium bromide cycle is a possible route to staying in the liquid region; another is to relax the temperature boost requirements. The outcome of raising the cooling-water temperature in the condenser by 10°C is shown in Case 2. The cycle can reach the 155°C value for T_{25} by increasing the heat exchanger UA values by 65% over those required for sodium hydroxide.

Table 1. Comparison of performance with different fluids for high waste heat temperatures

	Case 1		Case 2	
	Sodium hydroxide	Lithium bromide	Sodium hydroxide	Lithium bromide
x_1 , %	64.2	68.0	65.8	65.8
T_1 , °C	114.0	88.0	120.3	111.7
T_8 , °C	29.0	27.0	29.0	50.2
T_9 , °C	79.0	70.0	79.0	
T_{22} , °C	100.0	100.0	100.0	100.0
T_{23} , °C	90.0	90.0	90.0	90.0
T_{24} , °C	115.0	115.0	125.0	135.0
T_{25} , °C	135.0	135.0	155.0	155.0
T_{26} , °C	22.0	22.0	22.0	30.0
T_{27} , °C	25.0	25.0	25.0	35.0
T_{28} , °C	100.0	100.0	100.0	100.0
T_{29} , °C	90.0	90.0	90.0	90.0
Boost, ΔT , °C	35.0	35.0	55.0	55.0
$UAAO$, kW/K	43.0	700.0	245.0	900.0
$UAAR$, kW/K	147.0	228.0	223.0	573.0
$UAGI$, kW/K	69.0	72.0	67.0	154.0
$UAGR$, kW/K	594.0	25.0	239.0	43.0
UAE , kW/K	157.0	100.0	158.0	488.0
UAC , kW/K	458.0	700.0	458.	138.0
ΣUA , kW/K	1471.0	1826.0	1390.0	2296.0
COP	0.42	0.46	0.40	0.45
η , years	0.9	1.0	0.9	1.2
F	0.55	0.78	0.52	

The two case comparisons presented above show that indeed the same delivery temperatures may be reached with sodium hydroxide–water and lithium bromide–water solutions. In general, smaller temperature differences are required by lithium bromide–water, as reflected in the greater UA values for this fluid combination. Also, the condenser temperature must be raised for lithium bromide–water, which from a thermodynamic standpoint is undesirable.

The third case presented here is one in which sodium hydroxide solutions show a definite advantage over the lithium bromide–water ones. For waste heat temperatures lower than 100°C (as considered in Cases 1 and 2), the AGHX cycle with sodium

hydroxide offers much higher delivery temperature than lithium bromide–water. This is shown in Table 2, where a waste heat temperature of 60°C was assumed. Because of the crystallization-imposed limit, lithium bromide delivers process heat at a maximum temperature of 105°C. In sharp contrast, sodium hydroxide delivers at 123°C with basically the same total heat exchanger UA under the same conditions. Further increasing the recirculating loop UA values would increase this delivery temperature even more. Operation above 60% lithium bromide–water concentrations would require a dilution cycle before shutdown.

Table 2. Comparison of performance with different fluids
Case 3 for low waste heat temperatures

	Lithium bromide	Sodium hydroxide
$T_1, ^\circ\text{C}$	67.6	98.3
$T_2, ^\circ\text{C}$	107.7	125.8
$T_3, ^\circ\text{C}$	80.0	102.4
$T_4, ^\circ\text{C}$	66.0	69.4
$T_5, ^\circ\text{C}$	26.8	35.6
$T_6, ^\circ\text{C}$	44.0	60.3
$T_8, ^\circ\text{C}$	17.1	18.6
$T_9, ^\circ\text{C}$	53.9	52.4
$T_{22}, ^\circ\text{C}$	60.0	60.0
$T_{23}, ^\circ\text{C}$	50.0	50.0
$T_{24}, ^\circ\text{C}$	80.0	102.0
$T_{25}, ^\circ\text{C}$	105.0	123.2
$T_{26}, ^\circ\text{C}$	12.0	12.0
$T_{27}, ^\circ\text{C}$	15.0	15.0
$T_{28}, ^\circ\text{C}$	60.0	60.0
$T_{29}, ^\circ\text{C}$	55.0	55.0
$UAAO, \text{kW/m}^2$	700.0	400.0
$UAAR, \text{kW/m}^2$	526.0	450.0
$UAGI, \text{kW/m}^2$	84.2	250.0
$UAGR, \text{kW/m}^2$	57.6	600.0
$UAC, \text{kW/m}^2$	700.0	500.0
$UAE, \text{kW/m}^2$	700.0	500.0
$\Sigma UA, \text{kW/m}^2$	2768.0	2700.0
COP	0.40	0.41
$x_1, \%$	64.3	62.3
F	0.48	0.53

This example shows that the wider solution field of sodium hydroxide produces higher delivery temperatures than lithium bromide, provided that sodium hydroxide-water can be operated at the high concentrations. High concentrations can lead to practical problems, such as lack of vapor pressure equilibrium in the solution at the absorber. This low vapor pressure would considerably decrease the output temperature. Only experimental research with a prototype can answer these questions.

The third set of results focuses on the optimization problem. The optimizer may minimize the UA values needed to produce a specified output temperature T_{25} . However, this tends to decrease the COP considerably. As an example, the same cycle was calculated without and with the optimizer, reducing the sum of the UAs by nearly 50%. The COP was greatly reduced, from 0.41 to 0.04. This drop in COP may be corrected by imposing a lower limit for the UAs .

Perhaps the most important application of the optimizer is in improving the economics of the cycle. The output of this cycle can be valued in terms of the price of steam. As the heat exchanger area increases, the value of the output increases too. However, the capital outlay also increases for increasing heat exchanger area. Thus, there is a trade-off between initial investment and output value. The optimizer is then employed to find out the "best" trade-off. In our case, simple payback was employed as the figure of merit of the trade-off. More complete descriptions of economic feasibility such as internal rate of return or present worth can also be employed. The objective here is merely to show the technique whereby the optimizer can be used for the design of the heat pump with the best economics. It is recognized that more descriptive and complete economic techniques can be employed to obtain different solutions to this problem.

In the example, the payback period for the cycle described by the external temperatures in Table 3 is calculated as per Eq. 62. For the cycle shown in Fig. 6, the results are shown in Fig. 14, which displays the payback period as a function of the heat exchanger costs. The paybacks are short, even for the high price of \$480/m² of heat exchangers.

Table 3. Operating conditions of payback example

	Temperature (°C)
T_{22}	85.0
T_{23}	60.0
T_{24}	110.0
T_{25}	125.0
T_{26}	20.0
T_{27}	23.0
T_{28}	90.0
T_{29}	85.0

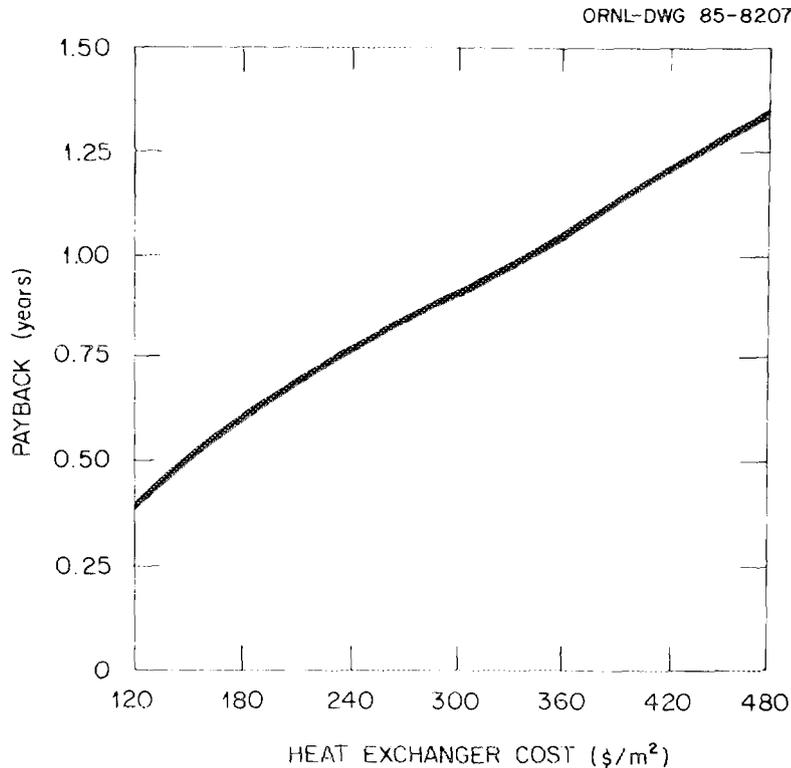


Fig. 14. Payback periods for the cycle shown in Fig. 6.

Figure 15 shows the payback for the same conditions for the cycle shown in Fig. 7. The payback periods tend to be larger, since the heat pump must provide heat to a cooler stream at T_{22} (Fig. 7) instead of T_{24} . Although this analysis is approximate, it does show that (1) if relatively low-cost heat exchangers (such as copper-nickel) are used, (2) if the waste heat is essentially free, and (3) if the output is valued at full steam price, then a temperature booster of this type can achieve low payback periods. In a recent study,¹⁵ evidence supporting the possibility of using sodium hydroxide and copper-nickel heat exchangers was presented.

The results presented so far assume that the external fluid in the output section of the absorber does not undergo a phase change, i.e., $T_{25} > T_{24}$. It is important to determine the potential of this cycle to produce steam. For the cycle shown in Fig. 6, it will be assumed that condensate at T_{24} is available and that it is necessary to have steam at T_{24} . The smallest temperature difference across the output heat exchanger will then be $T_{24} - T_3$, whereas before it usually was $T_{25} - T_2$. Therefore, although the output temperature is T_{24} , the cycle internal maximum temperature T_2 is much greater than T_{24} . Thus, the COP tends to be lower for a given output temperature for the steam case as compared with the sensible heat case.

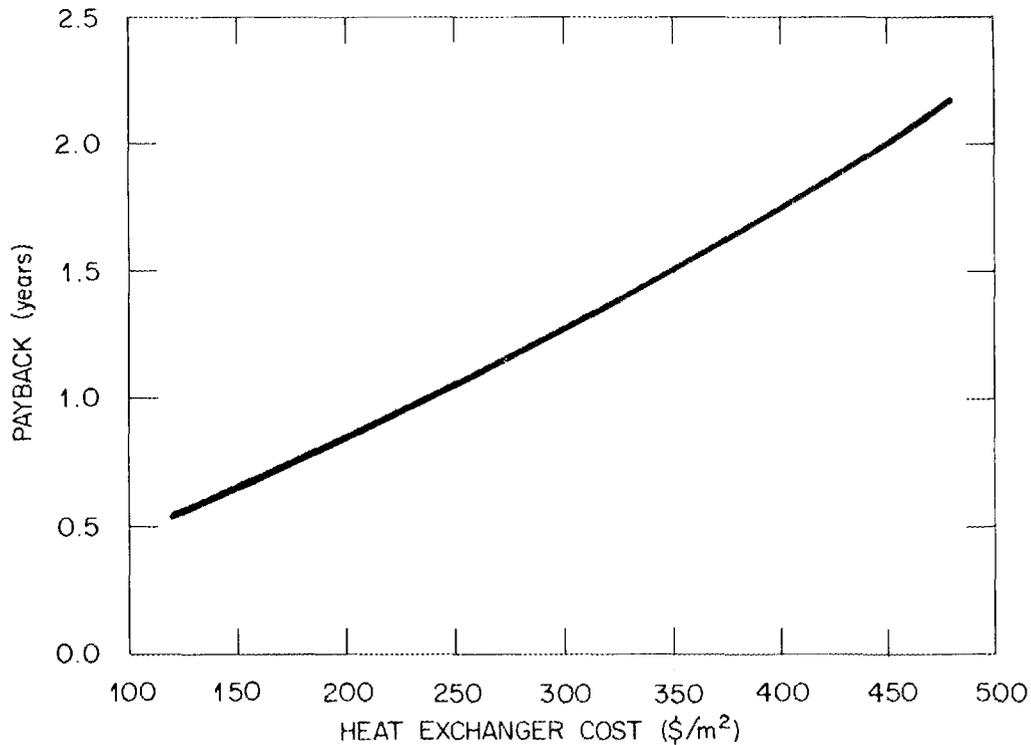


Fig. 15. Payback periods for the cycle shown in Fig. 7.

The results for the steam-producing case for the cycle shown in Fig. 6 are shown in Figs. 16 and 17. Figure 16 displays the optimized payback period vs the steam temperature for two different heat exchanger costs for the case presented in Table 1, with T_{24} variable. Comparison with Fig. 14 at an output temperature of 115°C shows that the payback period for producing steam is larger than that for sensible heat. However, even for the case of high heat exchanger costs, the payback falls in the 2-to-3-year range for output temperatures of 130 to 140°C . The COP shown in Fig. 17 decreases with output temperature.

For the case shown in Fig. 7, it is assumed that the water is preheated from T_{22} to T_{24} in the lower part of the absorber and that steam is produced in the upper part. The payback and COP optimized values are given in Figs. 18 and 19. A comparison of Figs. 16 and 18 shows that the payback periods of the latter tend to be longer than those of the former, especially so at high steam temperatures. This is reasonable because more heat per kilogram of steam must be delivered in the absorber in the case shown in Fig. 7 than in that shown in Fig. 6. The COP values of Fig. 19 are on the same order of those of Fig. 17, indicating that the effect of the extra heat delivered in the absorber on the COP is outweighed by additional heat supplied to the generator.

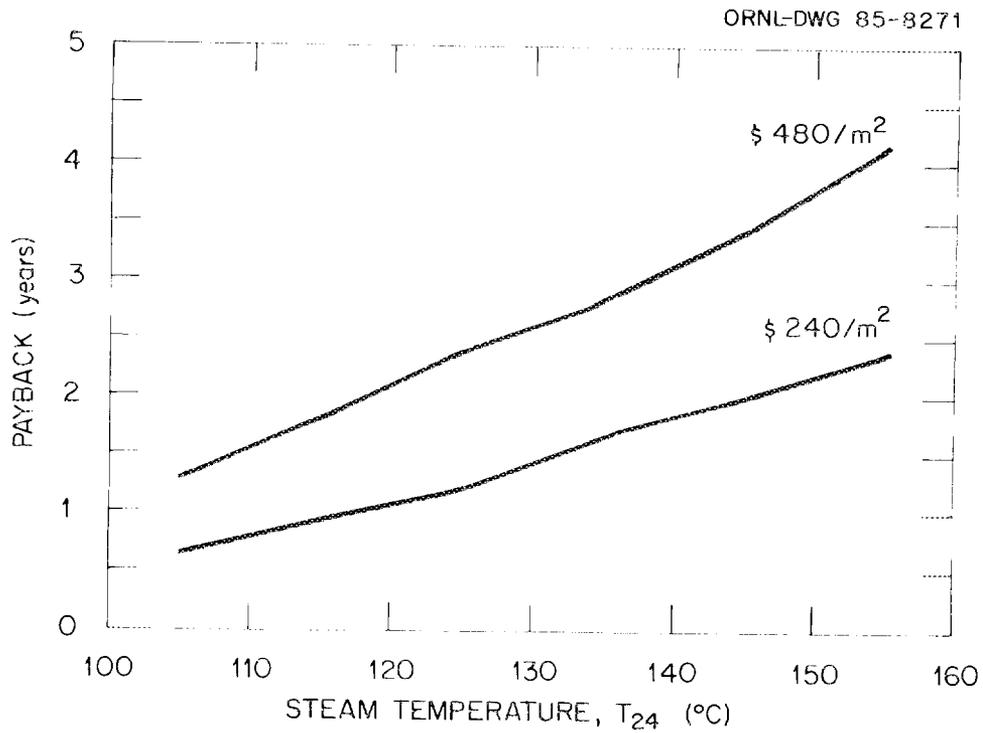


Fig. 16. Payback periods for the cycle shown in Fig. 6, steam-producing case.

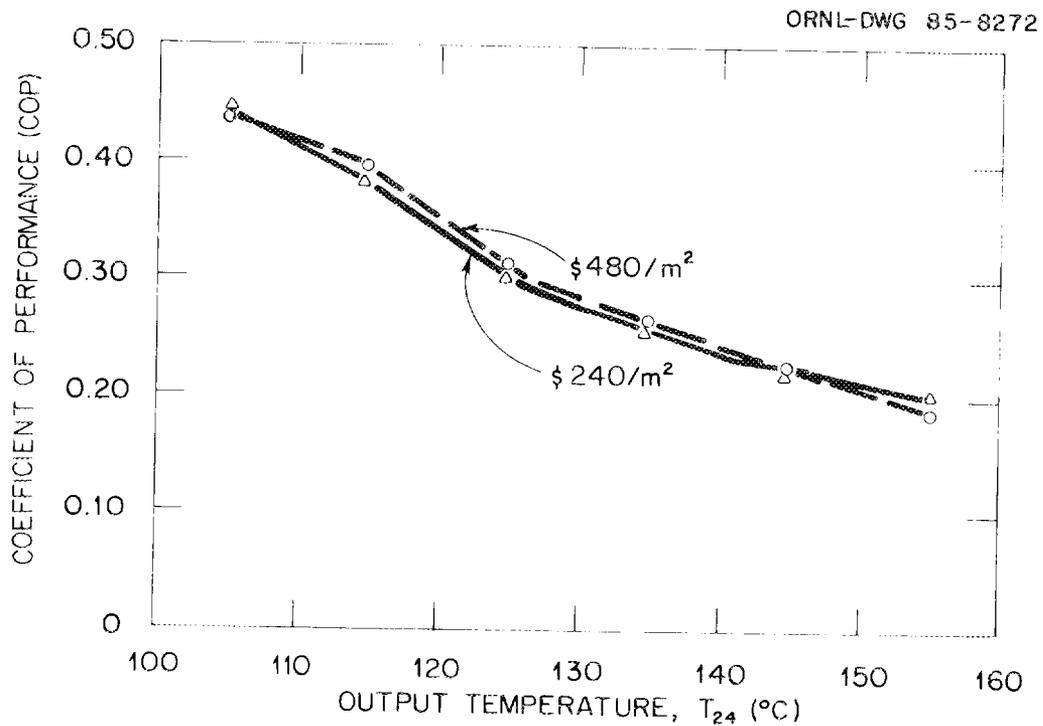


Fig. 17. COP values for the cases studied in Fig. 16.

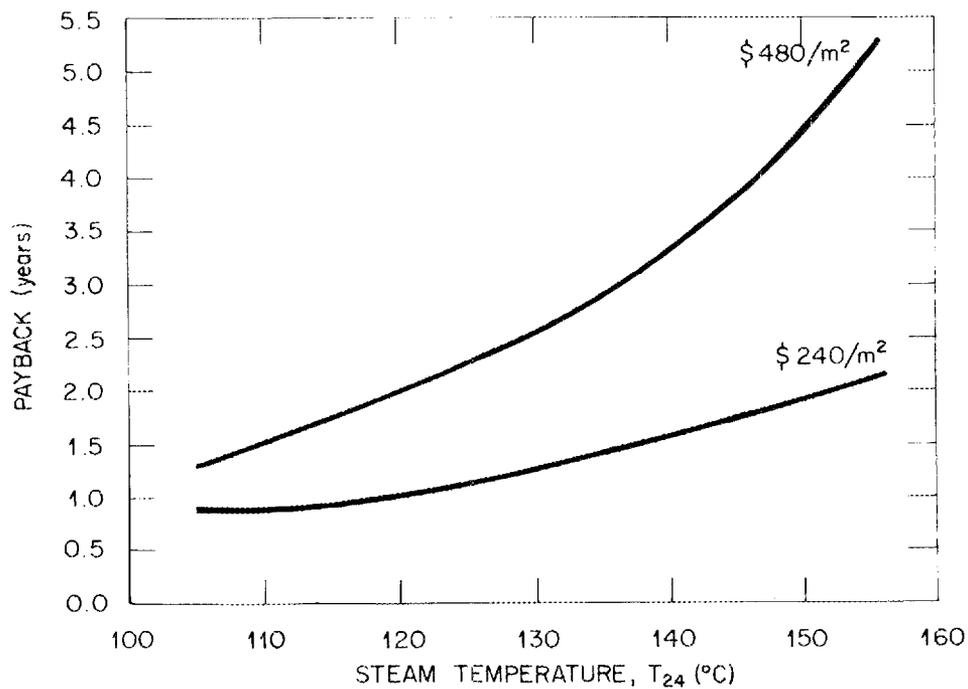


Fig. 18. Payback periods for the cycle shown in Fig. 7, steam-producing case.

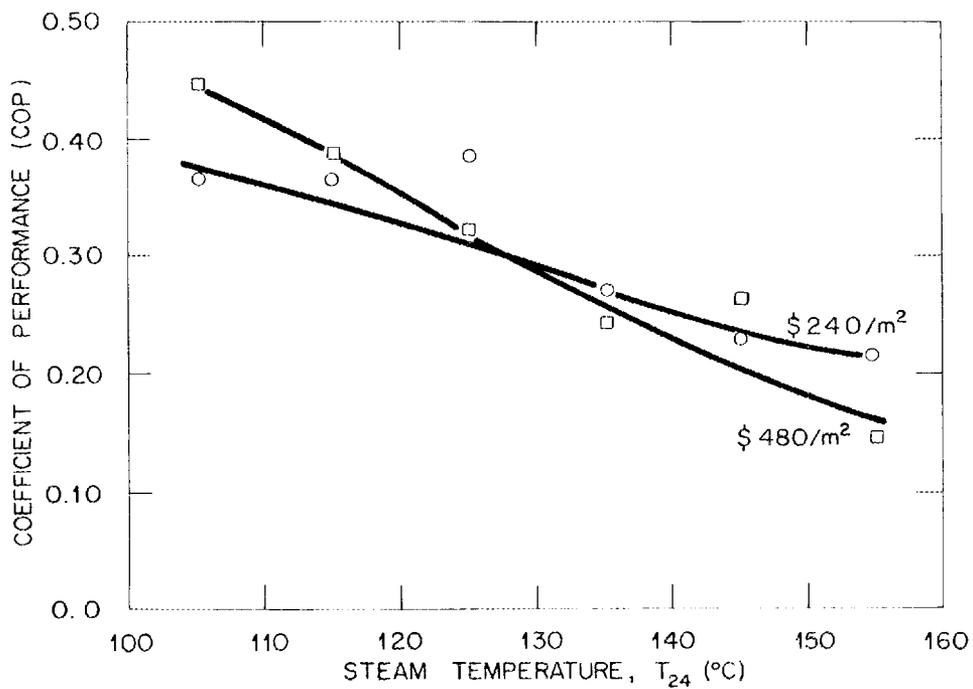


Fig. 19. COP values for the cases studied in Fig. 18.

5. CONCLUSIONS

In this report, the potential of a regenerative cycle operating with either sodium hydroxide-water or lithium bromide-water was explored. A numerical model of the cycle was formulated and implemented in a computer. An optimizer was coupled to the numerical model to produce optimal configurations satisfying different criteria.

The most significant conclusion of this work is that it is indeed possible to eliminate much of the design guesswork associated with the regenerative cycle by employing optimization techniques. It is shown how the sum of the transfer areas can be minimized for a given set of input and output temperatures. Alternatively, it is shown how the payback period can be minimized. The application of optimization methods for the design of absorption cycles is a new area. The present work is an incursion into this area, which appears very promising for the development of design tools.

There is some uncertainty in the price per square meter of one of these machines. This price would depend on materials of construction, manufacturing techniques employed, capacity, and number of machines actually made. However, even for relatively high prices (\$480/m²), this study shows the possibility of relatively short payback periods.

The regenerative cycle implemented with sodium hydroxide appears quite promising. Output temperatures of 180°C with waste heat temperatures of 100°C are shown to be possible. At waste heat temperatures of 100°C and with low condenser temperatures, sodium hydroxide shows the potential for larger temperature boosts than lithium bromide. At higher condenser temperatures, sodium hydroxide requires lower *UA* values than lithium bromide to reach the same delivery temperature.

Finally, the regenerative cycle is shown to greatly increase the flexibility of heat transformers over previous designs. For a given set of input waste heat and condenser temperatures, the cycle can produce a wide range of output temperatures, depending on the regenerative heat exchanger areas. The wider the operating field of the working fluids, the wider the range of delivery temperatures that the cycle can achieve.

In summary, coupling of optimization techniques with design of heat exchangers appears to be a fruitful field of endeavor. Further work will make the optimization routines more streamlined and adaptable to smaller computers that can be used at the designer's desk. The answers determined here in comparing sodium hydroxide-water systems with lithium bromide-water systems should be tested in prototype machines to establish their applicability.

REFERENCES

1. E. Miura, "Incentive Programs for Heat Pumps in Japan," *International Energy Agency Newsletter* 2(3), October 1984.
2. G. Grossman and K. W. Childs, "Computer Simulation of a Lithium Bromide-Water Absorption Heat Pump for Temperature Boosting," *ASHRAE Trans.* 89(1B), 240-248 (1983).
3. W. T. Hanna, M. L. Lane, and L. T. Whitney, "Industrial Applications for Waste Heat Powered Temperature Booster," pp. 1906-1910, *Proceedings of the Eighteenth Intersociety Energy Conversion Engineering Conference, Orlando, Florida, Aug. 21-26, 1983*, Vol. 4, American Institute of Chemical Engineers, 1983.
4. W. R. Huntley, "Performance Test Results of an Absorption Heat Pump That Uses Low-Temperature [60°C (140°F)] Industrial Waste Heat," pp. 1921-1926, *Proceedings of the Eighteenth Intersociety Energy Conversion Engineering Conference Orlando, Florida, Aug. 21-26, 1983*, Vol. 4, American Institute of Chemical Engineers, 1983.
5. H. Perez-Blanco, "Working Mixtures for Industrial Chemical Heat Pumps," pp. 21-34, *Second International Symposium on the Large Scale Applications of Heat Pumps, York, England, Sept. 25-27, 1984*, BHRA, Fluid Engineering Centre, 1984.
6. N. Isshiki, "Study of the Concentration Difference Energy System," *J. Non-Equilib. Thermodyn.* 2, 85-107 (1977).
7. H. Perez-Blanco and G. Grossman, *Cycle and Performance Analysis of Absorption Heat Pumps for Waste Heat Utilization*, ORNL/TM-7852, Oak Ridge National Laboratory, September 1981.
8. Anonymous, "Melco Develops Innovative Absorption Temperature Booster," *JARN* 16(9), 12 (Sept. 25, 1984).
9. G. Alefeld, "Rules for the Design of Multistage Absorption Machines," *Brenst-Waerme-Kraft* 34(2), 64-73 (1982).
10. B. A. Phillips, "Analyses of Advanced Residential Absorption Heat Pump Cycles," pp. 265-287, *Proceedings of the DOE/ORNL Heat Pump Conference, Research and Development on Heat Pumps for Space Conditioning Applications, Washington, D.C., Dec. 11-13, 1984*, CONF-841231, Oak Ridge National Laboratory, August 1985.
11. M. T. Heath (ed.), *Corlib Documentation*, Oak Ridge National Laboratory Computer Library, 1980.
12. Numerical Algorithm Group, *NAG Fortran Mini Manual*, Downer's Grove, IL 60516, 1980.
13. L. A. McNeely, "Thermodynamic Properties of Lithium Bromide," *ASHRAE Trans.* 85(1), 413-433 (1979).

14. R. H. Perry and C. H. Chilton, *Chemical Engineer's Handbook*, 5th Ed., McGraw-Hill, 1973.
15. J. C. Griess, J. H. DeVan, and H. Perez-Blanco, *Corrosion of Materials in Absorption Heating and Refrigeration Fluids*, ORNL/TM-9646, Oak Ridge National Laboratory, June 1985.

Appendix A

FIT OF ENTHALPY AND PARTIAL PRESSURE FOR SODIUM HYDROXIDE-WATER SOLUTION

A.1 ENTHALPY FIT

Data for both the enthalpy and partial pressure of sodium hydroxide-water were taken from the *Chemical Engineer's Handbook*.¹ The phase diagram for enthalpy was digitized using a Tektronix Model 4953 Graphics Tablet coupled to a PDP-10 computer. Approximately nine points were sampled from each isothermal line at intervals of 50 degrees starting at 50°C. In this manner, approximately 75 points were gathered for the temperatures of 50, 100, 150, 200, 250, 300, 350, and 400°C. For each of these points the enthalpy, concentration, and temperature were recorded in a file to be used for the fitting process.

Because of the strong linear dependence of enthalpy on temperature, the following form was assumed for the dependence of enthalpy (h) on concentration (x) and temperature (T):

$$h(x,T) = \sum_{i=0}^N a_i x^i + T \sum_{i=0}^N b_i x^i + T^2 \sum_{i=0}^N c_i x^i ,$$

where N is the highest power of x used in the series. Values of N as high as 5 were tried, along with setting all values of c_i equal to zero such that the equation is linear in temperature rather than quadratic. The goodness of fit was measured by a generalized standard deviation σ :

$$\sigma^2 = \sum_{j=1}^J [h_j - h(x_j, y_j)]^2 / (J - K) ,$$

where J = total number of points and K = total number of terms in the equation, i.e., 10 if $N = 4$ and all c_i values = 0. The coefficients were determined using the linear least squares routine LINLSQ from the computer library CORLIB² at ORNL. This routine uses Householder transformations to reduce the least squares matrix to upper triangular form, followed by back substitution. The value of N that was finally selected is 4, and the quadratic terms in T were dropped because the smallest value of σ was obtained in this fashion. The coefficients are given in the function subprogram F1(TIN,XIN) in the source listing for NAUOPT in Appendix B. The function F1 yields the enthalpy in kilojoules per kilogram given temperature TIN in degrees Celcius and concentration XIN in weight percent.

A.2 VAPOR PRESSURE FIT

Data for the vapor pressure as a function of concentration and temperature were digitized in a like manner to that for the enthalpy except that eight to ten points were obtained along lines of constant concentration. The values of vapor pressure were generated for concentration values of 0, 4.7, 9.1, 16.7, 23.0, 28.6, 33.3, 37.5, 41.2, 44.4, 47.4, 50.0, 54.5, 58.3, 61.5, 64.3, 66.7, 75.0, 77.8, 80.0, 83.3, and 87.5%. Fewer points were digitized for the higher concentration values because of the smaller length of the lines of constant concentration. The temperature range covered was from 0 to 250°C. At higher concentrations, the lower temperature was necessarily more than zero degrees. At the other extreme, no temperatures greater than 250°C were used in the fit.

Approximately 160 points were obtained for the pressure, and these points were fit to the equation

$$p_j = \exp \left(\sum_{k=0}^K \sum_{q=0}^Q c_{k,q} x_j^k T_j^q \right),$$

where j = data point number and N = number of terms in the expression for p , or $(K + 1)(Q + 1)$. The coefficients $c_{k,q}$ were determined using the same method that was used for the enthalpy coefficients: the routine LINLSQ was employed to solve the matrix equations. The results of this fit are given in the program listing of Appendix B in the function subprogram F2(TIN,XIN), which yields the pressure in pascals (N/m²) given temperature TIN in degrees Celcius and concentration XIN in weight percent.

A.3 REFERENCES

1. R. H. Perry and C. H. Chilton, *Chemical Engineer's Handbook*, 5th Ed., McGraw-Hill, 1973.
2. M. T. Heath (ed.), *Corlib Documentation*, Oak Ridge National Laboratory Computer Library, 1980.

Appendix B

LISTING AND EXAMPLES OF OUTPUTS OF THE PROGRAMS

B.1 LISTING OF FORTRAN CODE NAUOPT

```

00010 C          PROGRAM NAUOPT 6/19/85 - 2/10/86
00020 C          MODIFIED FROM MMOPT DATED 7/19/85
00030 C23456789112345678921234567893123456789412345678951234567896123456789712
00040 C
00050 C
00060 C          IMPLICIT REAL*8 (A-H,M,O-Z)
00070 C          EXTERNAL E04WAY
00080 C          INTEGER MEQ,MINEQ,MRNGE,M,MAXCAL
00085 C          LOGICAL LAMSET
00090 C          COMMON/BLOCK1/      MAR, MGA, MGI, T8, T9, X1, M1, MAO,
00100 C          1      UAGI,UAAR,UAGR,UAAO,UAC,UAE
00110 C          COMMON/BLOCK4/IPRINT,ITER
00120 C          COMMON /SCALE/XSCALE(15)
00130 C          EQUIVALENCE (X(1),MAR)
00140 C          NAMELIST/XMAIN/      MAR, MGA, MGI,T8,T9,X1,M1,MAO,
00150 C          1      UAGI,UAAR,UAGR,UAAO,UAC,UAE,FUNC,IFAIL
00160 C          NAMELIST/SUBARG/N,MEQ,MINEQ,MRNGE,M,
00170 C          1      IPRINT,MAXCAL,ETA,XTOL,STEPMX,CL,CU,LCLU,IBOUND,
00180 C          2      XL,XU,LAMSET,X,RHO,RLAM,FUNC,C,IW,LIW,LW,IFAIL
00190 C          DIMENSION X(15),Y(15),CL(1),CU(1),XL(15),XU(15),RLAM(15),C(15),
00200 C          1      IW(100),W(2000)
00210 C          FOR MINIMAX ROUTINE
00220 C          C          MAR      MGA      MGI      T8      T9      X1      M1      MAO
00230 C          C          UAAO      UAAR      UAGR      UAC      UAE      UAGI
00240 C          1      DATA XL/      0.0,      0.0,      0.0,      10.0,      50.0,      40.0,      0.0,      0.0,
00250 C          DATA XU/      10.0,      10.0,      25.0,      10.0,      10.0,      20.0,      0.0/
00260 C          1      DATA XU/      1.0,      1.0,      1.0,      50.0,      140.0,      75.0,      20.0,      1.0,
00270 C          1      700.0,      300.0,      700.0,      700.0,      700.0,      700.0,      0.0/
00280 C
00290 C**** OPEN(UNIT=10,ACCESS='SEQIN',FILE='FOR05.DAT')
00300 C          CALL ERRSET(208,256)
00310 C          CALL STAY
00320 C          NCASE=0
00330 C          110 IPRINT=0
00340 C          NCASE=NCASE+1
00350 C          ITER=1
00360 C**** READ(5,4,END=999)UAAO,UAAR,UAGR
00370 C          4  FORMAT(3F10.5)
00380 C          N=14
00390 C          NEW CASE NAEDEN FOR HFB 4/18/85
00400 C          STARTING VALUES ALTERED 1/16/85 TO RUN CASE FOR HFB
00410 C          NEW STARTING VALUES, 1/17/85, TO GET BETTER MINIMUM
00420 C          COMPARE TO LITHIUM BROMIDE RUN
00430 C          MAR=0.38
00440 C          MGA=0.21
00450 C          MGI=0.47
00460 C          T8=28.9
00470 C          T9=79.0
00480 C          X1=57.8
00490 C          M1=1.48
00500 C          MAO=0.56
00510 C          UAGI=57.3
00520 C          UAAR=201.4
00530 C          UAGR=59.4
00540 C          UAAO=323.1
00550 C          UAC=467.3
00560 C          UAE=158.0
00570 C**** ABOVE VALUES ARE NEAR THE EXPECTED RESULT, WE HOPE
00580 C**** FOR THE 3/1/85 11:45 NAUOPT RUN
00590 C          WRITE(6,1)N,IER,(X(I),I=1,N)
00600 C          1  FORMAT(' N=',I5,I5/' X =',(10F8.3))
00610 C          FOR E04UAF ROUTINE
00620 C          IF(NCASE.GT.1)GO TO 140
00630 C          DO 120 I=1,N
00640 C          XSCALE(I)=XU(I)*0.8
00650 C          X(I)=X(I)/XSCALE(I)
00660 C          XL(I)=XL(I)/XSCALE(I)
00670 C          120 CONTINUE
00680 C          DO 150 I=1,N
00690 C          XU(I)=XU(I)/XSCALE(I)
00700 C          150 CONTINUE
00710 C          MEQ=0
00720 C          MINEQ=15
00730 C          MRNGE=0
00740 C          M=MEQ+MINEQ+MRNGE
00750 C          IPRINT=10

```

```

00760      MAXCAL=4000
00770      ETA=0.01
00780      XTOL=0.10
00790      STEPMX=100000.0
00800      LCLU=1
00810      IBOUND=0
00820      LAMSET=.FALSE.
00830      RHO=10.0
00840      LIW=100
00850      LW=2000
00860      IFAIL=1
00870 C      28 ARGUMENTS TO E04UAF
00880      WRITE(6, SUBARG)
00890      CALL E04UAF(N, MEQ, MINEQ, MRNGE, M, E04WAY,
00900 1      IPRINT, MAXCAL, ETA, XTOL, STEPMX, CL, CU, LCLU, IBOUND,
00910 2      XL, XU, LAMSET, X, RHO, RLAM, FUNC, C, IW, LIW, W, LW, IFAIL)
00920 C      END MINIMAX ROUTINE CALL
00930      IPRINT=1
00940      CALL NAOHEQ(N, X, Y, IER)
00950      DO 130 I=1, N
00960      X(I)=X(I)*XSCALE(I)
00970 130      CONTINUE
00980      WRITE(6, XMAIN)
00990      WRITE(6, 1) N, IER, (X(I), I=1, N)
01000      WRITE(6, 2) (Y(I), I=1, 10)
01010 2      FORMAT(' Y= ', / (11F8.2))
01020      SUMSQ=0.0
01030      DO 100 I=1, N
01040      IF(I.GT.10) GO TO 100
01050      SUMSQ=SUMSQ+Y(I)**2
01060 100      CONTINUE
01070      SIG=DSQRT(SUMSQ/10.0D0)
01080      WRITE(6, 3) SIG
01090 3      FORMAT(' SIGMA= ', F10.2)
01100      WRITE(6, 5)
01110 5      FORMAT(' 0'/' 0')
01120 C*** 999      CLOSE(UNIT=10, ACCESS='SEQIN', FILE='FOR05.DAT')
01130 999      RETURN
01140      END
01150 C
01160 C
01170 C
01180      SUBROUTINE FUNCT1(IFLAG, N, X, FUN)
01190      IMPLICIT REAL*8 (A-H, O-Z)
01200      REAL*8 X(15), Y(15), XREAL(15)
01210      COMMON/SCALE/XSCALE(15)
01220      CALL NAOHEQ(N, X, Y, IDUM)
01230      FUN=0.0D0
01240      DO 100 I=1, N
01250      XREAL(I)=X(I)*XSCALE(I)
01260      IF(I.GT.10) GO TO 100
01270      FUN=FUN+Y(I)**2
01280 100      CONTINUE
01290      FUN=DSQRT(FUN/10.0D0)
01300      WRITE(6, 1) (XREAL(I), I=1, N), FUN
01310 1      FORMAT(15F7.2, E13.4)
01320      RETURN
01330      END
01340 C
01350 C
01360 C
01370      SUBROUTINE CON1(IFLAG, N, M, XC, CC)
01380      IMPLICIT REAL*8 (A-H, M, O-Z)
01390      INTEGER M
01400      COMMON/BLOCK3/ MAR, MGA, MGI, T8, T9, X1, M1, MAO,
01410 1      UAGI, UAAR, UAGR, UAAO, UAC, UAF
01420 1      COMMON/NAFUN/M9, T1, T2, T3, T4, T5, T6, T20, T21, T22, T23, T24,
01430      T25, T26, T27, T28, T29
01440      COMMON/SCALE/XSCALE(15)
01450      COMMON/PASS/TPASS(15)
01460      DIMENSION XC(N), CC(M)
01470      DATA NCON/0/
01480      NCON=NCON+1
01490      CC(1)=(T2-T25)*100
01500      CC(2)=(M9-MAR-MAO)*100.
01510      CC(3)=(M9-MGA-MGI)*100.
01520      CC(4)=(T3-T24)*100.
01530      CC(5)=(T3-T20)*100.
01540      CC(6)=(T4-T21)*100.
01550      CC(7)=(T8-T26)*100.
01560      CC(8)=(T29-T9)*100.
01570      CC(9)=(T28-T9)*100.
01580      CC(10)=(T8-T27)*100.
01590      CC(11)=(T22-T6)*100.
01600      CC(12)=(T23-T5)*100.
01610      CC(13)=(T20-T1)*100.
01620      CC(14)=(T21-T6)*100.
01630      CC(15)=(T3-T24)*100.0
01640      IF(NCON.EQ.1) WRITE(6, 1) (CC(I), I=1, M)
01650 1      FORMAT(' CON1: CC= ', 5(1PE12.3))
01660      RETURN
01670      END

```

```

01680 C
01690 C
01700 C
01710 C
01720 C
01730 C
01740 C
01750 C
01760 C
01770 C
01780 C
01790 C
01800 C
01810 C
01820 C
01830 C
01840 C
01850 C
01860 C
01870 C
01880 C
01890 C
01900 C
01910 C
01920 C
01930 C
01940 C
01950 C
01960 C
01970 C
01980 C
01990 C
02000 C
02010 C
02020 C
02030 C
02040 C
02050 C
02060 C
02070 C
02080 C
02090 C
02100 C
02110 C
02120 C
02130 C
02140 C
02150 C
02160 C
02170 C
02180 C
02190 C
02200 C
02210 C
02220 C
02230 C
02240 C
02250 C
02260 C
02270 C
02280 C
02290 C
02300 C
02310 C
02320 C
02330 C
02340 C
02350 C
02360 C
02370 C
02380 C
02390 C
02400 C
02410 C
02420 C
02430 C
02440 C
02450 C
02460 C
02470 C
02480 C
02490 C
02500 C
02510 C
02520 C
02530 C
02540 C
02550 C
02560 C
02570 C
02580 C

SUBROUTINE AMONIT(N,M,X,F,C,NITER,NF,GLNORM,COND,POSDEF,RHO,RLAM)
  IMPLICIT REAL*8 (A-H,O-Z)
  LOGICAL POSDEF
  DIMENSION X(N),C(M),RLAM(M)
  WRITE(6,1)NITER,F,GLNORM,COND,RHO
  FORMAT(' AMONIT: NITER, F, GLNORM, COND, RHO='
  /10X,I5,4(1PE12.3))
  RETURN
END

SUBROUTINE NAOHEQ(N,X,Y,IER)
  EQUATIONS FOR NAOH HEAT EXCHANGER 4/27/84-2/10/86
  IMPLICIT REAL*8 (A-H,M,O-Z)
  COMMON/BLOCK1/ MARA, MGAA, MGIA, T8A, T9A, X1A, M1A, MAOA,
  UAGIA, UAARA, UAGRA, UAAOA, UACA, UAEEA
  COMMON/BLOCK3/ MAR, MGA, MGI, T8, T9, X1, M1, MAO,
  UAGI, UAAR, UAGR, UAAO, UAC, UAE
  COMMON/BLOCK4/IPRNT, ITER
  COMMON/NAFUN/M9, T1, T2, T3, T4, T5, T6, T20, T21, T22, T23, T24,
  T25, T26, T27, T28, T29
  COMMON /SCALE/XSCALE(15)
  REAL*8 X(48),Y(48),XDUM(48),YSCALE(11),XALT(14),INSTAL
  EQUIVALENCE (MAR,XDUM(1)),(MARA,XALT(1))
  NAMELIST/NAM1/ M1, M2, M3, M4, M5, M6, M8, M9,
  MAO, MAR, MAA, MGA, MGI, MGR,
  X1, X2, X3, X4, X5, X6,
  T1, T2, T3, T4, T5, T6, T25,
  H1, H2, H3, H4, H5, H6, H8, H9,
  HGR, HGI, HGA,
  P7, P9, T8, T9, T6P, QE,
  QAR, QGR, QAO, QE, QG1, QC
  T20, T21, T22, T23, T24, T26, T27, T28, T29,
  UAGI, UAAR, UAGR, UAAO, UAC, UAE, COP, ITER, PAYBAK
  FUP
  NAMELIST/NAM2/ MAR, MGA, MGI, T8, T9, X1, M1, MAO,
  UAGI, UAAR, UAGR, UAAO, UAC, UAE
  DATA YSCALE/9*10.0,0.1,0.05/,DELTA/3.0/,INSTAL/1.3/,OVRHD/1.18/

F5(TABS)=10**(6.21147+(-2886.373-337269.46/TABS)/TABS)*6894.76
TRABS(TC)=1.8*TC+32.0+459.72
YSCALE(4)=1.0
YSCALE(9)=1.0

DO 120 I=1,N
  XDUM(I)=X(I)*XSCALE(I)
CONTINUE
NPI=N+1
IF(NPI.GT.14)GO TO 150
DO 140 I=NPI,14
  XDUM(I)=XALT(I)
CONTINUE
CONTINUE
CONTINUE
C*** REFERENCE DATA
T22=100.0
T23=90.0
T24=100.0
T25=135.0
T26=22.0
T27=25.0
T28=100.0
T29=90.0
C*** END REFERENCE DATA
M9=1.0
C BEGIN CYCLE
P7=F5(TRABS(T8))
H8=F6(T8)
P9=F5(TRABS(T9))
H9=F4(T9)
MAA=M9-MAR-MAO
MGR=M9-MGA-MGI
C*** ABSORBER
C*** ADIABATIC SECTION
M2=M1+MAA
X2=M1*X1/M2
T2=F21(P9,X2)
H2=F1(T2,X2)
H1=(M2*H2-MAA*H9)/M1
C*** OUTPUT SECTION
M3=M2+MAO
X3=M2*X2/M3
M4=M3+MAR
X4=M3*X3/M4
M5=M4-MGA
X5=M4*X4/M5
M6=M5-MGI
X6=M5*X5/M6

```

```

02590      T1=F21(P7,X1)
02600      H1A=F1(T1,X1)
02610 C***      NOTE H1A CHANGED FROM F1(T1,X6) ON 7/18/84
02620      T3=F21(P9,X3)
02630      T20=T24
02640      H3=F1(T3,X3)
02650      QAO=UAAO*DTLN(T2,T3,T25,T24)
02660      DELTO=T25-T24
02670      MAOP=(QAO+M3*H3-M2*H2)/H9
02680 C***      RECYCLE SECTION
02690      T4=F21(P9,X4)
02700      H4=F1(T4,X4)
02710 C***      GENERATOR
02720 C***      ADIABATIC SECTION
02730      T5=F21(P7,X5)
02740      H5=F1(T5,X5)
02750      HGA=F3(P7,(T4+T5)/2.0)
02760      MGAP=(M4*H4-M5*H5)/HGA
02770 C      INPUT SECTION
02780      T6=F21(P7,X6)
02790 C      NOTE CHANGE IN T21 FROM T21=T6+DELTA
02800      T21=T22
02810      QAR=UAAR*DTLN(T3,T4,T20,T21)
02820      DELTR=T20-T21
02830      IF(DELTO.LT.1.0)GO TO 160
02840      FUP=QAO*DELTR/(QAR*DELTO)
02850      GO TO 170
02860      FUP=(QAO*4.19*DELTR)/(QAR*2450.0)
02870      MARP=(M4*H4+QAR-M3*H3)/H9
02880      T6P=-UAAR/UAGR*(T4+T3-T20-T21)-T1+T20+T21
02890      HG1=F3(P7,(T6+T5)/2.0)
02900      H6=F1(T6,X6)
02910      QGI=UAGI*DTLN(T22,T23,T6,T5)
02920      MGIP=(M5*H5+QGI-M6*H6)/HG1
02930 C      RECYCLING SECTION
02940      HGR=F3(P7,(T6+T1)/2.0)
02950      QGR=UAGR*DTLN(T20,T21,T1,T6)
02960      MGRP=(M6*H6+QGR-M1*H1)/HGR
02970 C      CONDENSER
02980      M8=M9
02990      QC=UAC*DTLN(T8,T8,T26,T27)
03000      MG1A=(M8*H8+QC-MGR*HGR-MGA*HGA)/HG1
03010 C      EVAPORATOR
03020      QE=UAE*DTLN(T29,T28,T9,T9)
03030      M9P=(M8*H8+QE)/H9
03040      COP=(QAO+QAR*FUP)/(QGI+QE)
03050 C      SET UP TRIAL FUNCTION FOR E04UAF
03060      Y(1)=MAR*H9-(M4*H4+QAR-M3*H3)
03070      Y(2)=MGA*HGA-(M4*H4-M5*H5)
03080      Y(3)=MG1*HG1-(M5*H5+QGI-M6*H6)
03090      Y(4)=QAR*(1.0-FUP)-QGR
03100      Y(5)=H1A-H1
03110      Y(6)=MG1*HG1-(M8*H8+QC-MGR*HGR-MGA*HGA)
03120      Y(7)=M9*H9-(M8*H8+QE)
03130      Y(8)=MGR*HGR-(M6*H6+QGR-M1*H1)
03140      Y(9)=MAO*H9-(QAO+M3*H3-M2*H2)
03150      CAPITL=240.0*(UAAO+UAAR+UAGR+UAC+UAE+UAGJ)/2.0*INSTAL*OVRHD
03160      CAPITL=1.00*CAPITL
03170      SAVE=(QAO+QAR*FUP)*4.0D-6*3600.0*8760.0
03180 C      Y(10) IS THE PAYBACK IN YEARS
03190      PAYBAK=CAPITL/SAVE
03200      Y(10)=PAYBAK
03210 C***      Y(10)=1.0-COP
03220 C***      Y(10)=UAGI+UAAR+UAGR+UAAO+UAC+UAE
03230      SUMSQ=0.0
03240      DO 110 I=1,N
03250      IF(I.GT.10)GO TO 110
03260      Y(I)=Y(I)/YSCALE(I)
03270      SUMSQ=SUMSQ+Y(I)**2
03280      110 CONTINUE
03290      SIG=DSQRT(SUMSQ/10.0D0)
03300      IF (IPRNT.EQ.1) GO TO 130
03310      IPRINT=0
03320      IF(MOD(ITER-1,100).EQ.0)IPRINT=1
03330      IF(IPRINT.NE.1)GO TO 999
03340      130 WRITE(6,NAM1)
03350      WRITE(6,NAM2)
03360      WRITE(6,1)(XDUM(I),I=1,N)
03370      1 FORMAT(10A40,' NAOHEQ WAS CALLED, X='/(10F8.2))
03380      WRITE(6,2)(Y(I),I=1,10)
03390      2 FORMAT(10F8.2,' Y='/(11F8.2))
03400      WRITE(6,3)SIG
03410      3 FORMAT(' SIGMA=',F10.2/'0')
03420      999 ITER=ITER+1
03430      RETURN
03440      END
03450 C
03460 C
03470 C

```

```

03480      FUNCTION F1(TIN,XIN)
03490      C      ENTHALPY OF NAOH SOLN (KJ/KG) 4/03/84
03500      C      TIN=SOLN TEMP(DEG C), XIN=PERCENT NAOH
03510      C      FROM 10 TERM FIT 3/13/84 ENTHAL AKA TEMP
03520      IMPLICIT REAL*8 (A-H,O-Z)
03530      REAL*8 A(5),B(5)
03540      DATA
03550      1      A/ -3.518335D1,1.240801D2,-1.431403D3,4.749323D3,-3.167554D3/,
03560      2      B/ 1.036595D0,-1.500582D0,5.929570D0,-1.170812D1,7.320021D0/,
03570      3      KUP/5/,ITER/0/
03580      C      CONVERT LOCALLY TO FRACTIONAL CONCENTRATION AND DEG F
03590      X=XIN/100.0
03600      T=1.8*TIN+32.0
03610      C
03620      F1=0.0D0
03630      XK=1.0D0
03640      DO 100 K=1,KUP
03650      IF (K.GT.1) XK=X*XK
03660      F1=F1+XK*(A(K)+B(K)*T)
03670      C 100      CONTINUE
03680      C      CONVERT TO SI UNITS
03690      F1=F1*2.326009
03700      ITER=ITER+1
03710      C***      IF(ITER.LE.50)WRITE(6,1)TIN,XIN,F1
03720      1      FORMAT(' F1: TIN,XIN,F1='/'
03730      1      ' ,3E12.4)
03740      RETURN
03750      END
03760      C
03770      C
03780      C
03790      FUNCTION F2(TIN,XIN)
03800      C      PARTIAL PRESSURE OF WATER VAPOR IN PASCALS (N/M**2)
03810      C      OVER NAOH SOLUTION AS A
03820      C      FUNCTION OF TEMPERATURE (DEG C) AND CONCENTRATION (%)
03830      C      SEE PROGRAM PREFIT 3/29/84 13:43
03840      IMPLICIT REAL*8 (A-H,O-Z)
03850      DIMENSION A(5),B(5),C(5),D(5)
03860      DATA KUP/5/,ITER/0/,
03870      1      A/1.572342D0,-2.307659D-2,1.353628D-3,-6.090060D-5,
03880      2      3.525440D-7/,
03890      3      B/6.773735D-2,6.784010D-4,-5.987347D-5,1.798313D-6,
03900      4      -1.572486D-8/,
03910      5      C/-2.001039D-4,-4.270978D-6,3.681102D-7,-1.030200D-8,
03920      6      9.830376D-11/,
03930      7      D/2.723471D-7,8.807236D-9,-7.191766D-10,1.886063D-11,
03940      8      -1.837021D-13/
03950      XK=1.0D0
03960      F2=0.0D0
03970      DO 100 I=1,KUP
03980      IF (I.GT.1)XK=XK*XIN
03990      F2=F2+XK*(A(I)+TIN*(B(I)+TIN*(C(I)+TIN*D(I))))
04000      C 100      CONTINUE
04010      IF(F2.LT.170.0)GO TO 110
04020      F2=1.0E30
04030      GO TO 120
04040      110      F2=DEXP(F2)
04050      120      ITER=ITER+1
04060      C***      IF(ITER.LE.50)WRITE(6,1)TIN,XIN,F2
04070      1      FORMAT(' F2: TIN,XIN,F2='/'5X,3E12.4)
04080      F2=F2*133.322
04090      RETURN
04100      END
04110      C
04120      C
04130      C
04140      FUNCTION F2I(P,X)
04150      C      THIS FUNCTION YIELDS SOLUTION TEMP F2I (DEG C)
04160      C      CORRESPONDING TO THE PRESSURE P AND CONC X FOR NAOH SOLN
04170      IMPLICIT REAL*8 (A-H,O-Z)
04180      COMMON/DOG/PDOG,XDOG
04190      EXTERNAL TPX
04200      DATA ITER/0/
04210      PDOG=P
04220      XDOG=X
04230      C***      F2I=ZEROIN(0.0D0,250.0D0,TPX,0.05D0)
04240      IFAIL=1
04250      CALL C05ADF(0.0,250.0,1.0E-12,1.0E-13,TPX,F2I,IFAIL)
04260      ITER=ITER+1
04270      C***      IF(ITER.LE.50) WRITE(6,1)P,X,F2I
04280      1      FORMAT(' F2I: P,X,F2I=' ,3E12.4)
04290      RETURN
04300      END
04310      C
04320      C
04330      C
04340      FUNCTION TPX(T)
04350      C      THIS FUNCTION IS ZERO WHEN THE PARAMETER P (N/M**2)
04360      C      CORRESPONDS TO
04370      C      TEMPERATURE T (DEG C) AND CONCENTRATION X (%) FOR
04380      C      WATER VAPOR OVER AN NAOH SOLUTION
04390      IMPLICIT REAL*8(A-H,O-Z)

```

```

04400      COMMON/DOG/P,X
04410      TPX=P-F2(T,X)
04420      RETURN
04430      END
04440      C
04450      C
04460      C
04470      C
04480      FUNCTION F3(P,T)
04490      C      ENTHALPY OF SATURATED OR SUPERHEATED WATER VAPOR (STEAM)
04500      C      IN KILOJOULES/KILOGRAM (KJ/KG)
04510      IMPLICIT REAL*8 (A-H,O-Z)
04520      TF=1.8*T+32.0
04530      PSI=P/6894.76
04540      H=(0.00274*TF-0.989805)*PSI+0.44942*TF+1060.8
04550      F3=H*2.326009
04560      RETURN
04570      END
04580      C
04590      C
04600      C
04610      FUNCTION F4(TC)
04620      C      ENTHALPY OF SATURATED WATER VAPOR (KJ/KG)
04630      IMPLICIT REAL*8 (A-H,O-Z)
04640      F5(TABS)=10*(6.21147+(-2886.373-337269.46/TABS)/TABS)*6894.76
04650      TRABS(TC)=1.8*TC+32.0+459.72
04660      C
04670      PVS=F5(TRABS(TC))
04680      F4=F3(PVS,TC)
04690      RETURN
04700      END
04710      C
04720      C
04730      C
04740      FUNCTION F6(T)
04750      C      ENTHALPY OF SATURATED LIQUID WATER (KJ/KG)
04760      IMPLICIT REAL*8 (A-H,O-Z)
04770      TF=1.8*T+32.0
04780      HF=1.001*TF-32.05
04790      F6=HF*2.326009
04800      RETURN
04810      END
04820      C
04830      C
04840      C
04850      FUNCTION DTLN(T1,T2,T3,T4)
04860      C      LOGARITHMIC TEMPERATURE DIFFERENCE
04870      IMPLICIT REAL*8(A-H,O-Z)
04880      COMMON/BLOCK4/IPRINT,ITER
04890      DATA NOCALL/0/
04900      ARG1=T1-T3
04910      ARG2=T2-T4
04920      IF(ITER.LT.20)GO TO 110
04930      IF(ARG1*ARG2.LT.0.0)GO TO 130
04940      IF(ARG2.EQ.0.0)GO TO 100
04950      IF(ARG1.EQ.ARG2)GO TO 120
140      C***      DTLN=(ARG1-ARG2)/DLOG(ARG1/ARG2)
04960      C***      DTLN=(ARG1+ARG2)/2.0
04970      GO TO 999
04980      C***
04990      C*** 100      IF(IPRINT.LE.0)GO TO 110
05000      100      NOCALL=NOCALL+1
05010      WRITE(6,1)ITER,T1,T2,T3,T4
05020      1      FORMAT(' DTLN',I5,4E15.6)
05030      110      DTLN=(ARG1+ARG2)/2.0
05040      GO TO 999
05050      120      DTLN=ARG1
05060      GO TO 999
05070      130      X=ARG1
05080      Y=ARG2
05090      C***      IF(ARG1.GT.0.0.AND.ARG2.LT.0.0)X=-ARG1
05100      C***      IF(ARG1.LT.0.0.AND.ARG2.GT.0.0)Y=-ARG2
05110      C***      ARG1=X
05120      C***      ARG2=Y
05130      NOCALL=NOCALL+1
05140      WRITE(6,1)ITER,T1,T2,T3,T4
05150      C***      DTLN=X*Y
05160      C***      GO TO 140
05170      999      DTLN=(X+Y)/2.0
05180      RETURN
05190      END

```

B.2 NAZERO OUTPUT

ITERATION 1

N= 9 IER= 69
X =
0.54 0.19 0.39 27.10 105.50 162.60 69.50 0.89 0.41
\$NAM1
M1= 0.8900000000000000D+00, M2= 0.9400000000000000D+00, M3= 0.1350000000000000D+01, M4= 0.1890000000000000D+01,
M5= 0.1700000000000000D+01, M6= 0.1310000000000000D+01, M8= 0.1000000000000000D+01, M9= 0.1000000000000000D+01,
MAO= 0.4100000000000000D+00, MAR= 0.5400000000000000D+00, MAA= 0.5000000000000000D-01, MGA= 0.1900000000000000D+00,
MGI= 0.3900000000000000D+00, MGR= 0.4200000000000000D+00, X1= 0.6950000000000000D+02, X2= 0.658031914893617021D+02,
X3= 0.458185185185185185D+02, X4= 0.327275132275132275D+02, X5= 0.363852941176470588D+02, X6= 0.472175572519083969D+02,
T1= 0.131698300133454849D+03, T2= 0.184497999858134979D+03, T3= 0.142641868650868916D+03, T4= 0.124972167955124174D+03,
T5= 0.486506339985607047D+02, T6= 0.672059124902506074D+02, T25= 0.1626000000000000D+03, H1= 0.890389373033482068D+03,
H2= 0.985771167931454659D+03, H3= 0.620110051450635255D+03, H4= 0.469225266458858156D+03, H5= 0.218996234771706713D+03,
H6= 0.386863094984628397D+03, H8= 0.113534313577020000D+03, H9= 0.268356711711536677D+04, HGR= 0.268751587247074976D+04,
HGI= 0.260913485042638753D+04, HGA= 0.266365508385062434D+04, P7= 0.359187072070877209D+04, P9= 0.122881133720656063D+06,
T8= 0.2710000000000000D+02, T9= 0.1055000000000000D+03, T6P= -0.337200061199824445D+03, QE= -0.1400000000000000D+05,
QAR= 0.963754402665923990D+04, QGR= -0.472605315592636403D+03, QAO= 0.322699342545019474D+04, QE= -0.1400000000000000D+05,
QGI= 0.482869070223773759D+03, QC= 0.2880000000000000D+04, T20= 0.1000000000000000D+03, T21= 0.8000000000000000D+02,
T22= 0.8000000000000000D+02, T23= 0.6000000000000000D+02, T24= 0.1600000000000000D+03, T26= 0.1500000000000000D+02,
T27= 0.2000000000000000D+02, T28= 0.8000000000000000D+02, T29= 0.7500000000000000D+02, UAAO= 0.1000000000000000D+03,
UAAR= 0.2200000000000000D+03, UAC= 0.3000000000000000D+03, UAE= 0.5000000000000000D+03, UAGI= 0.4000000000000000D+02,
UAGR= 0.5000000000000000D+02, COP= -0.238733607169670070D+00, ITER= 1, \$
\$NAM2
MAR= 0.5400000000000000D+00, MGA= 0.1900000000000000D+00, MGI= 0.3900000000000000D+00, T8= 0.2710000000000000D+02,
T9= 0.1055000000000000D+03, T25= 0.1626000000000000D+03, X1= 0.6950000000000000D+02, M1= 0.8900000000000000D+00,
MAO= 0.4100000000000000D+00, \$
NAOHEQ WAS CALLED, X=
0.54 0.19 0.39 27.10 105.50 162.60 69.50 0.89 0.41
Y=
3.07 0.00 -0.2610110.15 7.19 0.13 -6.17 -0.70 0.76
SIGMA= 3370.05

ITERATION 82, CONVERGENCE

\$NAM3
MAR= 0.567013117913826268D+00, MGA= 0.266821679731798997D+00, MGI= 0.225272308207591458D+00, MGR= 0.0000000000000000D+00,
T8= 0.261015730371027631D+02, T9= 0.720517315081049114D+02, T25= 0.926144765535507831D+02, X1= 0.463421326700727502D+02,
M1= 0.358551210447174887D+01, MAO= 0.239566732251360937D+00, \$
N= 9 IER= 2
X =
0.57 0.27 0.23 26.10 72.05 92.61 46.34 3.59 0.24
Y=
-0.00 0.00 0.00 -0.02 0.00 0.00 -0.00 0.00 -0.00
SIGMA= 0.01
\$NAM1
M1= 0.358551210447174887D+01, M2= 0.377893225430656166D+01, M3= 0.401849898655792260D+01, M4= 0.458551210447174887D+01,
M5= 0.431869042473994987D+01, M6= 0.409341811653235842D+01, M8= 0.1000000000000000D+01, M9= 0.1000000000000000D+01,
MAO= 0.239566732251360937D+00, MAR= 0.567013117913826268D+00, MAA= 0.193420149834812794D+00, MGA= 0.266821679731798997D+00,
MGI= 0.225272308207591458D+00, MGR= 0.507906012060609545D+00, X1= 0.463421326700727502D+02, X2= 0.439701657647398423D+02,
X3= 0.413488414931535035D+02, X4= 0.362359260754199235D+02, X5= 0.384746905413084526D+02, X6= 0.405920609391205440D+02,
T1= 0.643586558737605633D+02, T2= 0.106758791058512901D+03, T3= 0.102460776424031101D+03, T4= 0.950615797035064820D+02,
T5= 0.506087840053799760D+02, T6= 0.539347574964220296D+02, T25= 0.926144765535507831D+02, H1= 0.367145403444943359D+03,
H2= 0.482976233847296178D+03, H3= 0.444731560291409529D+03, H4= 0.381266504644620793D+03, H5= 0.241881237433569000D+03,
H6= 0.271799857588198536D+03, H8= 0.109349903164062275D+03, H9= 0.263018191450332500D+04, HGR= 0.261147701398585645D+04,
HGI= 0.259850212561024061D+04, HGA= 0.263731091833854454D+04, P7= 0.338679962044707361D+04, P9= 0.340196192118128007D+05,
T8= 0.261015730371027631D+02, T9= 0.720517315081049114D+02, T6P= 0.385429771630740710D+02, QE= 0.252083200683741141D+04,
QAR= 0.153017736614822140D+04, QGR= 0.153020000561709170D+04, QAO= 0.668075180905794345D+03, QE= 0.252083200683741141D+04,
QGI= 0.653353063106133184D+03, QC= 0.250609733288186270D+04, T20= 0.1000000000000000D+03, T21= 0.8000000000000000D+02,
T22= 0.8000000000000000D+02, T23= 0.6000000000000000D+02, T24= 0.1600000000000000D+03, T26= 0.1500000000000000D+02,
T27= 0.2000000000000000D+02, T28= 0.8000000000000000D+02, T29= 0.7500000000000000D+02, UAAO= 0.1000000000000000D+03,
UAAR= 0.2200000000000000D+03, UAC= 0.3000000000000000D+03, UAE= 0.5000000000000000D+03, UAGI= 0.4000000000000000D+02

UAGR= 0.5000000000000000D+02, COP= 0.210471401693561804D+00, ITER= 83, \$
\$NAM2
MAR= 0.567013117913826268D+00, MGA= 0.266821679731798997D+00, MGI= 0.225272308207591458D+00, T8= 0.261015730371027631D+02,
T9= 0.720517315081049114D+02, T25= 0.926144765535507831D+02, X1= 0.463421326700727502D+02, M1= 0.358551210447174887D+01,
MAO= 0.239566732251360937D+00, \$
NAOHEQ WAS CALLED, X=
0.57 0.27 0.23 26.10 72.05 92.61 46.34 3.59 0.24
Y=
-0.00 0.00 0.00 -0.02 0.00 0.00 -0.00 0.00 -0.00
SIGMA= 0.01

NAOHEQ WAS CALLED. X=
0.39 0.16 0.52 29.16 79.13 57.61 1.49 0.55 45.26 234.75
58.93 308.00 443.41 158.77
Y=
-0.03 -0.04 0.03 -0.02 -0.99 0.19 0.07 0.33 -0.06 11.65
SIGMA= 3.70

&XMAIN
MAR= 0.391587720361891559 .MGA= 0.161094321313181840 .MGI= 0.418707105034916732 .T8= 29.1555197262312475 .T9=
79.1336338184063059 .X1= 57.6063280812115579 .M1= 1.48710653399898907 .MAO= 0.545105181912569417 .UAGI=
45.2637550758515417 .UAAR= 234.750620890418936 .UAGR= 58.9287453005532953 .UAAO= 308.000124283066214 .UAC=
443.410463218020766 .UAE= 158.767142138953830 .FUNC= 3.69939423056147088 .IFAIL= 2
&END
N= 14 0
X=
0.392 0.161 0.419 29.156 79.134 57.606 1.487 0.545 45.264 234.751
58.929 308.000 443.410 158.767
Y=
-0.03 -0.04 0.03 -0.02 -0.99 0.19 0.07 0.33 -0.06 11.65
SIGMA= 3.70

INTERNAL DISTRIBUTION

- | | |
|-----------------------|---------------------------------|
| 1. M. R. Ally | 18. J. F. Martin |
| 2. V. D. Baxter | 19. S. H. McConathy |
| 3. R. S. Carlsmith | 20. H. A. McLain |
| 4. R. C. DeVault | 21. V. C. Mei |
| 5. N. Domingo | 22. J. W. Michel |
| 6. S. K. Fischer | 23-24. M. R. Patterson |
| 7. B. J. Frame | 25. C. K. Rice |
| 8. W. Fulkerson | 26. R. E. Saylor |
| 9. L. D. Gilliam | 27. R. A. Stevens |
| 10. W. R. Huntley | 28. C. D. West |
| 11. M. R. Ives | 29. T. J. Wilbanks |
| 12. D. R. James | 30-31. Central Research Library |
| 13. S. I. Kaplan | 32. Document Reference Section |
| 14. J. O. Kolb | 33. Laboratory Records-----RC |
| 15. M. A. Kuliasha | 34-35. Laboratory Records |
| 16. R. S. Loffman | 36. ORNL Patent Section |
| 17. F. C. Maienschein | |

EXTERNAL DISTRIBUTION

37. W. Bierman, 45 Foxcroft Drive, Fayetteville, NY 13066
38. F. W. Bawel, Preway, 810 East Franklin Street, P.O. Box 534, Evansville, IN 47701
39. J. M. Calm, Electric Power Research Institute, 3412 Hillview Avenue, P.O. Box 10412, Palo Alto, CA 94303
40. Jaime G. Carbonell, Associate Professor of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213
41. R. N. Chappell, Program Manager, Energy Conservation Branch, Energy and Technology Division, DOE Idaho Operations Office, 785 DOE Place, Idaho Falls, ID 83402
42. D. D. Colosimo, Mechanical Technology, Inc., 968 Albany-Shaker Road, Latham, NY 12110
43. D. A. Didion, National Bureau of Standards, Thermal Machinery Group, Building Equipment Division, CBT, Washington, DC 20234
44. P. D. Fairchild, Electric Power Research Institute, 3412 Hillview Avenue, P.O. Box 10412, Palo Alto, CA 94303
45. Mark Falek, Gas Energy Inc., 166 Montague Street, Brooklyn, NY 11201

46. R. J. Fiskum, Program Manager, Energy Conversion Equipment Branch, CE-132, Room GH-068, Department of Energy, 1000 Independence Avenue SW, Washington, DC 20585
47. S. Malcolm Gillis, Professor, Public Policy and Economics, Duke University, 4875 Duke Station, Durham, NC 27706
48. G. Grossman, Technion Institute of Technology, Faculty of Mechanical Engineering, Haifa, Israel
49. W. T. Hanna, Battelle Columbus Laboratories, 505 King Avenue, Columbus, OH 43201
50. F. C. Hayes, Trane Company, 3600 Pammel Creek Road, LaCrosse, WI 54601
51. Fritz R. Kalhammer, Vice President, Electric Power Research Institute, P.O. Box 10412, Palo Alto, CA 94303
52. J. L. Kamm, University of Toledo, Scott Park Campus, Toledo, OH 43606
53. R. E. Kasperson, Professor of Government and Geography, Graduate School of Geography, Clark University, Worcester, MA 01610
54. S. A. Klein, University of Wisconsin, Mechanical Engineering Department, Madison, WI 53706
55. M. Lessen, Consulting Engineer, 12 Country Club Drive, Rochester, NY 14618
56. R. Macriss, Institute of Gas Technology, 3424 South State Street, Chicago, IL 60616
57. A. R. Maret, Gas Research Institute, 8600 West Bryn Mawr Avenue, Chicago, IL 60631
58. L. A. McNeely, 7310 Steinmeier Drive, Indianapolis, IN 46250
59. K. P. Murphy, Allied Corporation, Morristown, NY 07960
60. Horacio Perez-Blanco, Phillips Engineering Company, 721 Pleasant Street, St. Joseph, MI 49085
61. B. A. Phillips, Phillips Engineering Company, 721 Pleasant Street, St. Joseph, MI 49085
62. R. Radermacher, University of Maryland, Mechanical Engineering Department, College Park, MD 20742
63. Will Rebello, PAR Enterprises, 11928 Appling Valley Road, Fairfax, VA 22030
64. R. C. Reimann, 5504 Ortloff Road, LaFayette, NY 13084
65. S. L. Richlen, Office of Industrial Programs, CE-141, FORSTL, Department of Energy, 1000 Independence Avenue SW, Washington, DC 20585
66. J. D. Ryan, Energy Conversion Equipment Branch, CE-132, Room GH-063, Department of Energy, 1000 Independence Avenue SW, Washington, DC 20585
67. J. Saunders, Brookhaven National Laboratory, Upton, NY 11973
68. G. C. Vliet, Taylor Hall 116, The University of Texas, Austin, TX 78712
69. M. Wahlig, Lawrence Berkeley Laboratory, University of California, Berkeley, CA 94720
- 70-79. Energy Conservation Distribution, Oak Ridge National Laboratory, Building 9102-2, Room 103, Oak Ridge, TN 37831

- 80. Office of the Assistant Manager for Energy R&D, U.S. Department of Energy, Oak Ridge Operations, Oak Ridge, TN 37831
- 81-106. Technical Information Center, Oak Ridge, TN 37831