Multiple Solutions in Absorption Refrigeration Systems

James Gates
Advisors: Dr. S. Paolucci and Dr. M. Sen

Last Updated: July 27, 2007

Abstract

Absorption refrigeration is an old technology, but by using ionic liquids instead of traditional absorbents, we add a new spin to the AR cycle. Ionic liquids are more environmentally friendly and non-corrosive, therefore making them good alternatives to Freon, chlorine-based refrigerants, ammonia, and lithium bromide. The absorption refrigeration cycle can be modeled by a large set of nonlinear equations. Solving for a steady state of the cycle involves solving this system using numerical methods. It has been suggested that because of nonlinearities present, multiple steady state solutions may exist. Using iterative numerical methods, a single steady state is calculated, and which steady state the system converges to depends on initial conditions given. A numerical MATLAB code has been created that takes a system of nonlinear equations and produces the set of all known steady state solutions. This code, or a code adapted from this original one can be used to solve for multiple steady states in the AR cycle if real, physically possible multiple solutions exist and proper initial conditions are given.
1 Introduction and Objectives

Absorption refrigeration relies on thermal energy input to power the cycle. Four major components, along with heat exchangers, expansion valves, and a pump, make up the system, as shown schematically in Figure (1) Using mass, momentum, and energy balances on all

![Diagram of the Absorption Refrigeration Cycle]

Figure 1: Absorption Refrigeration Cycle

components, the behavior of the system is mathematically modeled. The result is a large nonlinear system of equations. Because of the nonlinearities, multiple solutions may exist. This research has attempted to isolate multiple steady state solutions to the absorption refrigeration cycle, and hopefully has brought us closer to finding these solutions.

A complete listing of the equations describing the cycle behavior can be found in [1]. These equations have been reduced in an attempt to simplify the system. The work and advancements made during this research project can be found in the file entitled WeeklyUpdates.pdf, available on my personal web page. This outlines how the equations were reduced and goes into more detail about the different approaches made in attempting to solve for the multiple solutions. Without further ado, I will continue on with a general overview of what was covered in this summer’s research project.
2 Attempted Solution Methods

Although the sponsored project is to investigate the use of ionic liquids in the absorption refrigeration cycle, a proper equation of state for the ionic liquids has not yet been derived, and the thermal and transport properties of the liquid are unknown at this time. These properties lie at the heart of finding the steady state solution to the cycle, so the well-known ammonia-water system was analyzed. At least one solution to this cycle is known, so any methods employed in an attempt to find solutions of a cycle using ionic liquids and carbon dioxide can be tested against the same calculations performed for the ammonia-water cycle to see if the known solution for ammonia-water arises.

2.1 Trying to Find Known Solution

As mentioned above, the ammonia-water cycle has been in use for almost a century. It’s behavior has been extensively studied, and the steady state solution is known. It must first be attempted to find this known solution in an attempt to familiarize oneself with the solution process. Once the general approach has been developed and verified, certain parameters can be easily adjusted so that the model created can be applied to a cycle using ionic liquids as an absorbent.

To solve for the known solution, several unknown values were assumed to be constant, such as heat transfer and mass flow rates, among others. This can be done using excessive amounts of writing and calculator calculations, or a simple program can be created that models the steady state behavior and transient response, as has been created by Cai in [1]. These programs, however, have a certain drawback. While they work great for telling us what the known steady state solutions is, they rely heavily on assuming certain values as chosen constants, such as mass flow rate of the refrigerant. Such models seek the steady state solution based on numerical substitution of equations, and therefore cannot be used to find multiple steady state solutions, where full equation substitution must be used, not merely the numerical values calculated at some given constants.

2.2 Solving a Purely Algebraic System

In order to solve for multiple steady state solutions, all equations defining the system’s behavior must remain analytical. These equations can then be substituted into one another, reducing the system to a more manageable number of equations. In the case of the AR cycle, this method was employed to reduce the system from 49 to 29 equations.

Some of the equations defining the cycle are equations for fluid enthalpies at various locations. The enthalpy depends on temperature, pressure, and mole fraction. It must be calculated from an appropriate equation of state. I have been unable to come up with a correct enthalpy equation for the ammonia-water solution, and have thus been unable to isolate multiple solutions. What I did find however, is that the enthalpy equations yield exponential components, and the system no longer is purely polynomial (all other equations are polynomials).

While solving a large system of nonlinear polynomial equations proves difficult enough in itself, solving this system when not all terms are polynomial becomes mathematically
impossible. An analytical approach must be abandoned in favor of a numerical method that looks for multiple steady-state solutions. Numerical calculations are done using the MATLAB programming software.

2.3 Numerical Solutions Using MATLAB

MATLAB contains an internal function called fsolve. This function takes a system of nonlinear equations and calculates one steady state state solution. It should be noted that not all solutions found using fsolve are steady state, as the system sometimes diverges for given initial conditions. For a diverging system, iterations within fsolve cease and an incorrect answer is returned. This false answer can be easily recognized using an option in fsolve which shows the deviation of the system from the steady state response. A shortcoming of this function is that for certain initial guesses, it cannot converge on a known solution. Fsolve is the most useful tool available, unless we want to write an entirely new code to accomplish almost the same task as fsolve. It has been found that approximately 0.1% of solutions returned by fsolve are erroneous.

Using the fsolve function and basic MATLAB programming, code can be written that runs through many initial guesses, and the processes the steady state results given by fsolve to determine if more than one unique solution exists. This has been accomplished using a small sample set of equations that I have created, and will be discussed in more detail below.

3 Solving a Nonlinear System

These next few sections describe how a nonlinear system can actually be solved numerically, and how multiple solutions can be obtained.

3.1 Example Problem

As an example problem, I created a set of five nonlinear equations in five unknowns. The problem was constructed in such a way that I first picked the solutions and then wrote the equations from the known solutions. In this case, the five equations had 12 possible solutions which must be found in MATLAB. If a program I have created successfully finds all 12 solutions to this example problem, I can conclude that it should be able to find multiple solutions the a similar problem with more unknowns, should multiple solutions exist.

I will direct your attention to the file called ExNonlinearSolver.m for the full code used to find solutions. In addition, the function myfun.m contains the five equations in five unknowns, and another program called ReducedSolutions post processes the data to remove all repeat solutions. These files can all be found at http://www.nd.edu/~jgates1/IonicLiquidsFiles-ND/test_problem/. In these codes you can see that nested loops take many combinations of initial guesses. These initial guesses are then sent the the fsolve function, along with a function containing the equations. Fslove uses an iterative process (Newton-Raphson Method) to optimize the system of equations, thus finding a steady state solution. This steady state solution is stored in a matrix of all solutions calculated. The solution depends on initial guesses, and as the initial guesses vary, fsolve will converge to different solutions.
Once the data has all been stored, it is post processed using a short code I have written. This code takes every solution and compares it to the others, eliminating all repeat solution. The output is a set of unique solutions. For the example problem, this gave all 12 known solutions, as well as 133 erroneous solutions. All solutions can be checked by plugging back into the original equations to see if they are correct. The 133 equations that were not solutions to the system can be easily eliminated using this method. In fact, a quick scan of the solutions will show the user which are correct due to extremely large or small numbers, but verification by the given equations never hurt anyone.

3.2 Absorption Refrigeration Cycle Problem

The absorption refrigeration cycle proves slightly more difficult than any small sample problem I have created. Multiple solutions may exist, but I don’t know what they are ahead of time, so I must be more careful in choosing initial guesses and in which solutions I dismiss as erroneous. Also, one must be extremely careful that the equations initially entered are correct and represent the conservation of mass, energy, and momentum in the AR cycle.

3.2.1 Defining Equations

Using mass, energy, and momentum balances, a complete set of equations describing the behavior of the cycle can be derived. Initially this was 49 equations, which I will not rewrite here. If interested, please see my Weekly Updates for the original 49 equations.

The system, however, has been reduced using basic algebra to 29 equations in 29 unknowns. Equation forms are not known for the enthalpies, so these must be determined before correct solutions to the system can be found. It has been suggested that tables for the enthalpies may also be used, but I am unsure how this would be accomplished. The equations that I have come up with are listed as follows:

\[
\begin{align*}
    h_1 &= h_1(P_G, \dot{V}_1, T_G, X = 1) \\
    h_3 &= h_3(P_C, \dot{V}_2, T_C, X = 1) \\
    h_5 &= h_5(P_A, \dot{V}_5, T_A, X_A) \\
    h_9 &= h_9(P_G, \dot{V}_9, T_A, X_G) \\
    K_1 &= \dot{m}_r(h_1 - h_3) \\
    K_2 &= \dot{m}_r, h_1 - \dot{m}_s, h_7 + (\dot{m}_s - \dot{m}_r)h_8 \\
    K_3 &= \dot{m}_r(h_1 - h_3) \\
    K_4 &= \dot{m}_r, h_4 + (\dot{m}_s - \dot{m}_r)h_9 - \dot{m}_s, h_5 \\
    K_5 &= \dot{m}_s(h_7 - h_6) \\
    K_5 &= (\dot{m}_s - \dot{m}_r)(h_8 - h_9)
\end{align*}
\]
\[
\dot{m}_s[(1 - X_G)(X_AM_r + (1 - X_A)M_o) - (1 - X_A)(X_GM_r + (1 - X_G)M_o)]
= \dot{m}_r[(1 - X_G)(X_AM_r + (1 - X_A)M_o)]
\]
\[
\Delta P \tilde{V}_5 = (h_6 - h_5)(X_AM_r + (1 - X_A)M_o)
\]
\[
\dot{m}_r | \dot{m}_r | \tilde{V}_1 = K_6(P_G - P_C)
\]
\[
\dot{m}_r | \dot{m}_r | \tilde{V}_2 = K_7(P_G - P_{V_1}^f)
\]
\[
\dot{m}_r | \dot{m}_r | \tilde{V}_3 = K_8(P_{V_1}^s - P_E)
\]
\[
\dot{m}_s | \dot{m}_s | \tilde{V}_4 = K_{10}(P_E - P_A)
\]
\[
\dot{m}_s | \dot{m}_s | \tilde{V}_5 = K_{11}(X_AM_r + (1 - X_A)M_o)(P_A - P_e^s + \Delta P)
\]
\[
\dot{m}_s | \dot{m}_s | \tilde{V}_6 = K_{12}(X_AM_r + (1 - X_A)M_o)(P_e^s - P_G)
\]
\[
(m_s - \dot{m}_r) \tilde{V}_9 = K_{13}(X_GM_r + (1 - X_G)M_o)(P_G - P_{V_2}^s)
\]
\[
(m_s - \dot{m}_r) \tilde{V}_9 = K_{14}(X_GM_r + (1 - X_G)M_o)(P_{V_2}^s - P_e^s)
\]
\[
(m_s - \dot{m}_r) \tilde{V}_{10} = K_{15}(X_GM_r + (1 - X_G)M_o)(P_{V_2}^s - P_A)
\]
\[
P_G[\tilde{V}_1^3 - \tilde{V}_1(b_1X_G^2 + b_2(1 - X_G)^2 + b_3(X_G(1 - X_G)))] =
RT_G\tilde{V}_1^2 + RT_G\tilde{V}_1(b_1X_G^2 + b_2(1 - X_G)^2 + b_3(X_G(1 - X_G)))
- \tilde{V}_1(a_1X_G^2 + a_2(1 - X_G)^2 + a_5(X_G(1 - X_G)))
+ (a_1X_G^2 + a_2(1 - X_G)^2 + a_5(X_G(1 - X_G)))*
(b_1X_G^2 + b_2(1 - X_G)^2 + b_3(X_G(1 - X_G)))
\]
\[
P_C(\tilde{V}_2^3 - \tilde{V}_2b_1^2) = RT_C\tilde{V}_2^2 + RT_C\tilde{V}_2b_1 - a_1\tilde{V}_2 + a_1b_1
\]
\[
P_E(\tilde{V}_4^3 - \tilde{V}_4b_1^2) = RT_E\tilde{V}_4^2 + RT_E\tilde{V}_4b_1 - a_1\tilde{V}_4 + a_1b_1
\]
\[
P_A[\tilde{V}_5^3 - \tilde{V}_5(b_1X_A^2 + b_2(1 - X_A)^2 + b_3(X_A(1 - X_A)))] =
RT_A\tilde{V}_5^2 + RT_A\tilde{V}_5(b_1X_A^2 + b_2(1 - X_A)^2 + b_3(X_A(1 - X_A)))
- \tilde{V}_5(a_1X_A^2 + a_2(1 - X_A)^2 + a_5(X_A(1 - X_A)))
+ (a_1X_A^2 + a_2(1 - X_A)^2 + a_5(X_A(1 - X_A)))*
(b_1X_A^2 + b_2(1 - X_A)^2 + b_3(X_A(1 - X_A)))
\]

In the above 29 equations, constants include \(K_{1-15}, \Delta P, M_o, M_r, b_{1-3}\), and \(a_{1,2,5}\). These constants all have definitions that I will not go into at this time, but are given in my MATLAB programs. This system of equations is sent to the fsolve function for optimization with all right-hand sides set equal to zero, and a solution is obtained.

### 3.2.2 Solution Methodologies

Multiple solutions can be isolated using MATLAB .m-files modeled after those used for the example problem. I have already created skeletons of such programs. These can be
found at http://www.nd.edu/~jgates1/IonicLiquidsFiles-ND/ARcycle_problem/. NonlinearSolver1.m sets initial conditions and calls the fsolve function. I also have lines of code that quickly calculate a coefficient of performance based on the results obtained from fsolve. MY own function, myfun1.m contains 23 complete equations and 6 unknown enthalpy equations. For the time being, I have assumed the enthalpies to be constant and have them entered in the equations matrix in this way. No code for post processing of the data has yet been created, however several updates must be made before the code can be run.

First I will point out that although I have checked my own equations over for accuracy, no one else has yet looked over them for errors in math of transcribing them into the computer codes. This should be done before they can be accepted as absolutely correct. Second, if the"brute force method" as performed in the example problem is used to find multiple solutions, a code will take up to five months to run in MATLAB on a windows machine. In this span of time, the likelihood that the computer will be shut down or the power go out is very likely. Problems may also arise in the solutions matrix becoming too large for the memory. Speed can be increased by suppressing all writing to the command window, which takes most of the time. I believe this can be done running a command line interface for MATLAB in UNIX. Alternatively, a similar program could be written in a faster language such as Fortran.

In any case, further work must be done before multiple solutions to the absorption refrigeration cycle can be proved, but recent work has advanced us toward accomplishing this goal. An outline of the steps to follow has been created, and the details must now be filled in appropriately.

4 Conclusions

During the past ten weeks, I have studied refrigeration systems and have gained a greater knowledge of their operation. I searched for multiple solutions to the AR cycle in hopes of finding an alternate steady state solution that is more efficient than the one currently used.

Dynamic modeling methods proved unsuccessful in trying to isolate multiple solutions. Likewise, when an algebraic approach was made, solutions could not be found due to the complexity and numerous nonlinearities present. Finally a numerical method using a large number of varying initial guesses was used. Solutions calculated using iterative procedures were compared against one I did stuff, but not nearly enough to make it worth my while. Yep. another and then compared back to the original equations to determine if solutions calculated were actually solutions and were unique.

This basic approach has been used to solve a smaller system of nonlinear differential equations, and is to be adapted to the larger system I have developed based on mass, energy, and momentum balances in the absorption refrigeration cycle.

Through the course of my work, I have gained a greater understanding of refrigeration cycles and ionic liquids. I was posed with a difficult problem, that of solving a nonlinear system for multiple steady states. While I was unable to produce definitive results, I have helped to develop a procedure that can be used to find these multiple solutions from the system of defining equations.
References