8-14-1992

Binary Mixtures Near Surfaces

Patrick Dunn
Portland State University

AN ABSTRACT OF THE THESIS OF Patrick Dunn for the Master of Science in Physics presented August 14, 1992.

Title: Binary Mixtures Near Surfaces

APPROVED BY THE MEMBERS OF THE THESIS COMMITTEE:

[redacted]

Erik Bodegom, Chair

[redacted]

Jonathan Abramson

[redacted]

David Paul

[redacted]

Gavin Bjork

This paper presents an approach to modeling a binary mixture near a surface. The approach used is based on statistical mechanics. The Cluster Variation (CV) method is used to solve an Ising-like model with a cluster size of two (neighboring lattice points). The free energy of the system is expressed in terms of the probabilities of occurrences of particular clusters. The CV method is first developed for a homogeneous system which leads to a set of three non-linear partial differential equations. The surface boundary is then introduced and the CV method is developed for this system which leads to a set of seven non-linear partial differential equations for each layer. In the calculations, the

Natural Iteration (NI) method is introduced as a method of solving these non-linear partial differential equations. The steps involved in implementing the NI method are outlined. The number of layers is chosen to be 32. Concentration profiles are calculated for the binary system using the NI method and the results are discussed.

BINARY MIXTURES NEAR SURFACES

by

PATRICK DUNN

A thesis submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE
in
PHYSICS

Portland State University
1992

TO THE OFFICE OF GRADUATE STUDIES:

The members of the Committee approve the thesis of Patrick Dunn presented August 14, 1992.

[redacted]

Erik Bodegom, Chair

[redacted]

Jonathan Abramson

[redacted]

David Paul

[redacted]

Gavin Bjork

APPROVED:

[redacted]

Erik Bodegom, Chair, Department of Physics

[redacted]

Roy W. Koch, Vice Provost for Graduate Studies and Research

TABLE OF CONTENTS

PAGE

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

CHAPTER I

INTRODUCTION

This study involved calculating concentration profiles for a binary mixture near a solid wall. Each molecule occupies a single site on a three dimensional simple cubic lattice. The binary mixture consists of two liquids that have, in the bulk phase, a region in the temperature-concentration space where they are not completely miscible. A classic example of such a mixture is water and nicotine. This behavior can be understood at a relatively simple level by assuming that there are repulsive interactions between the different types of molecules and an attraction between like molecules. This attraction or repulsion comes about because of the presence or absence of the particular intermolecular forces such as hydrogen bonding.

In this model, the exact nature of these forces does not play any role. Near walls, a concentration profile can develop depending on the interactions between the wall and the different types of molecules. These forces can give rise to the peculiar phenomena described in Beysens and Estere (1985) where one liquid wets the walls of a container in apparent defiance to the laws of gravity.

Different methods can be used to calculate the concentration profiles such as Bethe's Approximation, the Kramers-Wannier Approximation and the mean-field approximation (Kikuchi, 1951; Bodegom, 1982). This study, however, involved the Cluster Variation (CV) method (Kikuchi, 1977, 1979) which is a generalization of the aforementioned approximations and which is used to solve Ising-like models (Kikuchi, 1951, 1979). In the CV method, a maximum cluster size is chosen. The CV method is based on expressing the free energy of the system as a function of a set of variables which properly describe the system (i.e., the probabilities of occurrences of particular clusters). The "system," in this case, is composed of all the lattice points which, in turn, are occupied by two types of molecules—say A and B.

If the cluster is a single point (the simplest choice), one obtains the mean field approximation which is what DeOliveira and Griffiths (1978) used. This is an approximation that is relatively poor and is known to give completely erroneous results for a one-dimensional system. For a higher dimensional Ising model, such measures as the critical temperature and critical exponents deviate by large amounts from the correct values. A much better approximation is afforded by the pair approximation. There, the cluster size consists of two neighboring lattice points. This approximation is also known as the Bethe Approximation or the quasi-chemical approximation. Among the higher order and better approximations, one finds the Kramers-Wannier Approximation which corresponds to a cluster size of four.

This study approached the question of the wetting of the wall from the pair approximation. This step is one step better than, for instance, what DeOliveira and Griffiths did. Nevertheless, it was not so complicated that the problem could not be solved on a personal computer.

The CV method leads to a set of partial differential equations which must be solved simultaneously. The Natural Iteration (NI) method is then employed as a means of solving these equations. A computer program was written in FORTRAN to implement the NI method.

The concentration profiles were then calculated at various temperatures. The effects of raising the temperature on the already stable system were investigated.

CHAPTER II

THEORY

ISING MODEL

The Ising model assumes that at each lattice point there exists a small magnet called a spin magnetic moment and that this magnetic moment can point in either the positive or negative *z* direction. One then speaks of spin up or spin down. The model assumes that a bond of neighboring parallel spins (two spins of the same sign) contributes an amount of energy $-\varepsilon$ and a bond of antiparallel neighboring spins (two spins of the opposite sign) contributes an amount of energy $+\varepsilon$ to the system. A more stable configuration is obtained when the total energy is negative and, likewise, a less stable configuration is obtained when the total energy is positive.

For a collection of spins on a particular lattice, the stable configuration is obtained by minimizing the free energy. The problem is in calculating the entropy. An exact solution is only obtainable in certain simple cases. For a three dimensional lattice, one has to resort to approximate methods. A general approach to the problem of calculating the entropy in principle, to any accuracy, was introduced by Kikuchi (1951). This method is called the Cluster Variation (CV) method. In this method, one introduces the probabilities of occurrences of certain concentrations of spins or of molecules (in the case of a molecular system).

CLUSTER VARIATION METHOD

As an example, consider a linear lattice composed of M lattice points where each of the nearest neighbor lattice points form a bond as in the Ising model. Such a configuration can be considered to be a system and L such systems is considered to be an assembly as Figure 1 illustrates.

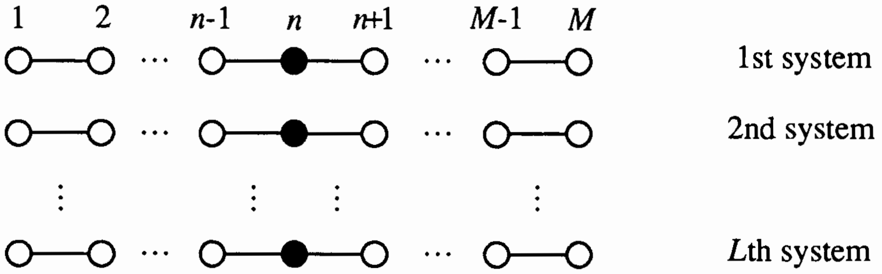

Figure 1. An assembly of linear lattices.

Looking at all of the nth lattice points of the assembly, all of the bonds formed between the molecules on the (n-1)th and nth lattice sites, and all of the bonds formed between the molecules on the nth and (n+1)th lattice sites, the fraction of each kind of bond in the assembly and the associated energy is shown in Table I. If there are L lattice points, $y_{11}L$ is the number of ++ bonds between the (n-1)th and nth sites and between the nth and (n+1)th sites, $y_{12}L$ is the number of +- bonds, etc., in the assembly. For a homogeneous system, the y_{ij} are independent of the lattice sites under consideration. The fractional variables are normalized as follows:

$$\sum_{i,j=1}^{2} y_{ij} = 1 \tag{2.1}$$

For a homogeneous system:

$$y_{12} = y_{21} \tag{2.2}$$

TABLE I

BOND CONFIGURATIONS

Bond Configuration	Fractional Variable y_{ij}	Associated Energy
$++$	y_{11}	$-\varepsilon$
$+-$	y_{12}	$+\varepsilon$
$-+$	y_{21}	$+\varepsilon$
$--$	y_{22}	$-\varepsilon$

One can also consider the configuration of a point. The concentration of spins up, x_1, follows from the definition of the y_{ij}:

$$x_1 = y_{11} + y_{12} \tag{2.3}$$

and, similarly, the concentration of spins down, x_2, is:

$$x_2 = y_{21} + y_{22} \tag{2.4}$$

and

$$x_1 + x_2 = 1 \tag{2.5}$$

The free energy of the system can be defined in terms of the internal energy E, the absolute temperature T, and the entropy S:

$$F = E - TS \tag{2.6}$$

Defining $\Phi M = \beta F$ and $\beta = 1/kT$ where k is Boltzman's constant and M is the number of lattice points in the system, equation (2.6) can be rewritten as:

$$\Phi M = \beta E - \frac{S}{k} \tag{2.7}$$

where S=$k$ln$G$, with G the number of ways a system that has the right distribution can be assembled. Φ is thus the free energy per lattice point. Although equations (2.1)-(2.5) were developed for a linear lattice, they are also valid for a three-dimensional lattice.

Rather than discussing the linear system, a simple cubic lattice will be considered. *E* and *S* must be defined in terms of the CV variables. Kikuchi (1951, 1977) has shown that:

$$\frac{E}{M}=3\sum_{i,j=1}^{2}\varepsilon_{ij}y_{ij} \tag{2.8}$$

and

$$\frac{S}{Mk}=5\sum_{i=1}^{2}x_i(\ln x_i-1)-3\sum_{i,j=1}^{2}y_{ij}(\ln y_{ij}-1) \tag{2.9}$$

where *S* is found by finding the number of ways, *G*, the system, can be assembled. Combining equations (2.7)-(2.9) leads to:

$$\Phi=3\beta\sum_{i,j=1}^{2}\varepsilon_{ij}y_{ij}-5\sum_{i=1}^{2}x_i(\ln x_i-1)+3\sum_{i,j=1}^{2}y_{ij}(\ln y_{ij}-1)+\beta\lambda(1-\sum_{i,j=1}^{2}y_{ij}) \tag{2.10}$$

where the last term of equation (2.10) is based on the constraint of equation (2.1) and λ is a Lagrange multiplier. Goldstein (1980) describes the use of Lagrange multipliers as a method of reducing the number of dependent variables. Taking the partial derivatives $\frac{\partial\Phi}{\partial y_{11}},\frac{\partial\Phi}{\partial y_{12}}$, and $\frac{\partial\Phi}{\partial y_{22}}$, because of equation (2.2) $\frac{\partial\Phi}{\partial y_{12}}=\frac{\partial\Phi}{\partial y_{21}}$ yields:

$$\frac{\partial\Phi}{\partial y_{11}}=-3\beta\varepsilon-5\ln x_1+3\ln y_{11}-\beta\lambda \tag{2.11}$$

$$\frac{\partial\Phi}{\partial y_{12}}=6\beta\varepsilon-5\ln x_1-5\ln x_2+6\ln y_{12}-2\beta\lambda \tag{2.12}$$

$$\frac{\partial\Phi}{\partial y_{22}}=-3\beta\varepsilon-5\ln x_2+3\ln y_{22}-\beta\lambda \tag{2.13}$$

Setting equations (2.11)-(2.13) equal to zero and solving for the y_{ij} yields the extremum values:

$$y_{11} = (x_1 x_1)^{5/6} e^{\beta\lambda} e^{\beta\varepsilon} \qquad (2.14)$$

$$y_{12} = (x_1 x_2)^{5/6} e^{\beta\lambda} e^{-\beta\varepsilon} \qquad (2.15)$$

$$y_{22} = (x_2 x_2)^{5/6} e^{\beta\lambda} e^{\beta\varepsilon} \qquad (2.16)$$

These equations can be solved using the Natural Iteration method which will be explained later. It should be emphasized that equations (2.14)-(2.16) are for a homogeneous system.

THE MODEL

Thus far, the concepts of spins or molecules were used interchangeably. From here on, only molecules are considered. The model that is the main item in this thesis is one in which two types of molecules (say *A* and *B*; indices on the fractional variables will still be 1 and 2) interact with a wall or boundary. The same configuration, as used by DeOliveira and Griffiths is studied. This configuration is one of a simple cubic lattice, with the boundary along a {1 0 0} plane and a finite number of layers extending perpendicular from the boundary. A finite number is required because of limits based on computational resources.

The surface introduced in the binary mixture will have an interaction with the molecules on the sites. This interaction can take on many possibilities. Only one case will be investigated. There the wall has an attractive interaction with one type of molecule (say *A*) and an equal in magnitude, but repulsive, interaction with the other type of molecule. Other interactions can be investigated with the same programs. In addition, from Pandit, et al. (1982), we know that this could give rise to interesting phenomena if *A* is the minority bulk phase. The particular phenomenon of interest is the occurrence of first order phase transitions in the excess surface density of *A* molecule.

Summarizing, we have a system with a wall that interacts with a binary ("liquid") mixture. The molecules occupy sites on a regular simple cubic lattice. Due to the interactions of the wall with the different types of molecules and the interaction of the molecules with each other, the concentrations near the wall can show drastic changes upon variations in temperature or bulk concentration. In order to find the equilibrium profiles of the concentration near the wall, the free energy for this non-homogeneous system needs to be calculated. This calculation is addressed in the next sections.

THE NON-HOMOGENEOUS SYSTEM

Since the system will no longer be homogeneous, equation (2.2) is no longer valid. Translational symmetry still exists along the two directions which are parallel to the surface boundary, but the direction perpendicular to the surface boundary no longer has this translational symmetry. In order to account for the interaction of the surface with the lattice points, modifications need to be made to the previously developed equations. Figure 2 illustrates a cross section of the system and indicates the layers which are parallel to a surface. If we take the y_{ij} to be in the direction perpendicular to the surface and look at the *n*th layer, there will be a local concentration of *A* and *B* molecules, x_1^n and x_2^n, and there will now be an x_1 to the right of the *n*th layer called $x_1^{n+1}$and an x_1 to the left of the *n*th layer called x_1^{n-1} and similarly there will be an x_2^{n+1} and x_2^{n-1}. With y_{ij}^n the probability of finding an *i* molecule in the *n*th layer and a *j* molecule in the (*n*+1)th layer, one can see that:

$$x_1^n = y_{11}^n + y_{12}^n \tag{2.17}$$

$$x_2^n = y_{21}^n + y_{22}^n \tag{2.18}$$

$$x_1^n = y_{11}^{n-1} + y_{21}^{n-1} \tag{2.19}$$

$$x_2^n = y_{12}^{n-1} + y_{22}^{n-1} \tag{2.20}$$

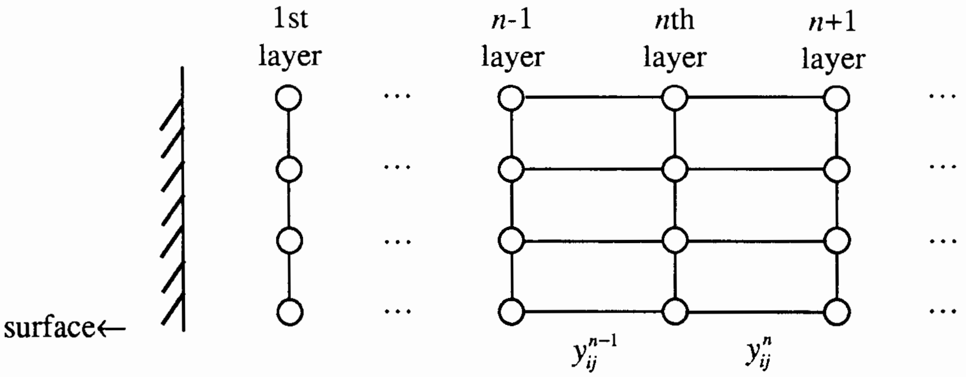

Figure 2. Surface boundary and cross section of layers.

Equations (2.17) and (2.19) should give the same concentrations. For convenience, the x_i^n in equation (2.17) and (2.18) will be labeled as x_i^{nR}, where R stands for right. This refers to the fact that in these equations the sum is over pair probabilities that relate to the bond on the right. Similarly, the x_i^n in equations (2.19) and (2.20) will be labeled x_i^{nL} for obvious reasons.

If the two directions parallel to the surface are defined by t_{ij} (analogous with the y_{ij}), then

$$x_1^{nt} = t_{11}^n + t_{12}^n \tag{2.21}$$

$$x_2^{nt} = t_{21}^n + t_{22}^n \tag{2.22}$$

$$t_{12}^n = t_{21}^n \tag{2.23}$$

where equations (2.21)-(2.23) are similar to those developed for the homogeneous system. We now have the constraints that

$$x_1^{nR} = x_1^{nL} = x_1^{nt} \tag{2.24}$$

$$x_2^{nR} = x_2^{nL} = x_2^{nt} \tag{2.25}$$

Equation (2.8) is now modified to represent the energy in the nth layer, E^n, as

$$E^n = \beta\left[2\sum_{i,j=1}^{2}\varepsilon_{ij}t_{ij}^n + \sum_{i,j=1}^{2}\varepsilon_{ij}y_{ij}^{n-1} + \sum_{i=1}^{2}\gamma_i^n x_i^n\right] \tag{2.26}$$

where γ_i^n represents the interaction of the x_i lattice point in the nth layer with the surface. Because of constraint equations (2.24) and (2.25), and the usefulness of having symmetric equations for the NI method, x_i^n is written as:

$$x_i^n = \sum_{i,j=1}^{2}\left[\frac{1}{6}y_{ij}^n + \frac{1}{6}y_{ji}^{n-1} + \frac{2}{3}t_{ij}^n\right] \tag{2.27}$$

Combining (2.26) and (2.27) yields:

$$E^n = \beta\left[2\sum_{i,j=1}^{2}\varepsilon_{ij}t_{ij}^n + \sum_{i,j=1}^{2}\varepsilon_{ij}y_{ij}^{n-1} + \sum_{i=1}^{2}\gamma_i^n\sum_{j=1}^{2}\left[\frac{1}{6}y_{ij}^n + \frac{1}{6}y_{ji}^{n-1} + \frac{2}{3}t_{ij}^n\right]\right] \tag{2.28}$$

Equation (2.9) is now modified to represent the entropy in the nth layer, S^n, as:

$$\frac{S^n}{k} = 5\sum_{i=1}^{2}x_i^n(\ln x_i^n - 1) - 2\sum_{i,j=1}^{2}t_{ij}^n(\ln t_{ij}^n - 1) - \sum_{i,j=1}^{2}y_{ij}^n(\ln y_{ij}^n - 1)$$

$$= \frac{10}{3}\sum_{i=1}^{2}x_i^{nt}(\ln x_i^{nt} - 1) + \frac{5}{6}\sum_{i=1}^{2}x_i^{nL}(\ln x_i^{nL} - 1) + \frac{5}{6}\sum_{i=1}^{2}x_i^{nR}(\ln x_i^{nR} - 1)$$

$$-2\sum_{i,j=1}^{2}t_{ij}^n(\ln t_{ij}^n - 1) - \sum_{i,j=1}^{n}y_{ij}^n(\ln y_{ij}^n - 1) \tag{2.29}$$

where the factors 10/3 and 5/6 are chosen to produce symmetric equations for the NI method. The other constraints on the system are as follows:

$$\sum_{i,j=1}^{2}y_{ij}^n = 1 \tag{2.30}$$

$$\sum_{i,j=1}^{n}t_{ij}^n = 1 \tag{2.31}$$

where equations (2.30)-(2.31) are the normalization conditions. Additionally one has for layer 2 to (N - 1): $x_1^{nR} - x_2^{nR} = x_1^{nL} - x_2^{nL}$ (2.32)

for layer 2 to (N - 1): $x_1^{nR} - x_2^{nR} + x_1^{nL} - x_2^{nL} = 2(x_1^{nt} - x_2^{nt})$ (2.33)

for 1st layer only: $x_1^{1R} - x_2^{1R} = x_1^{1t} - x_2^{1t}$ (2.34)

for last (Nth) layer only: $x_1^{N} - x_2^{N} = x_1^{NL} - x_2^{NL}$ (2.35)

where equations (2.32)-(2.35) are based on combinations of equations (2.24) and (2.25).

An "applied magnetic field-like" quantity H (or chemical potential) is introduced as a Lagrange multiplier to constrain the bulk concentration. Combining equations (2.28)-(2.35), and summing over N layers yields:

$$\Phi = \sum_{n=2}^{N-1}\Bigg[\beta\Bigg[2\sum_{i,j=1}^{2}\varepsilon_{ij}t_{ij}^{n} + \sum_{i,j=1}^{2}\varepsilon_{ij}y_{ij}^{n-1} + \sum_{i=1}^{2}\gamma_i\sum_{j=1}^{2}\left[\frac{1}{6}y_{ij}^{n} + \frac{1}{6}y_{ji}^{n-1} + \frac{2}{3}t_{ij}^{n}\right]\Bigg]$$

$$-\frac{10}{3}\sum_{i=1}^{2}x_i^{nt}(\ln x_i^{nt} - 1) - \frac{5}{6}\sum_{i=1}^{2}x_i^{nL}(\ln x_i^{nL} - 1) - \frac{5}{6}\sum_{i=1}^{2}x_i^{nR}(\ln x_i^{nR} - 1)$$

$$+2\sum_{i,j=1}^{2}t_{ij}^{n}(\ln t_{ij}^{n} - 1) + \sum_{i,j=1}^{2}y_{ij}^{n}(\ln y_{ij}^{n} - 1) + 2\beta\lambda_1^{n}\left(\sum_{i,j=1}^{2}t_{ij}^{n} - 1\right) + \beta\lambda_2^{n}\left(\sum_{i,j=1}^{2}y_{ij}^{n} - 1\right)$$

$$+\beta\mu^{n}(x_1^{nR} - x_2^{nR} - x_1^{nL} + x_2^{nL}) + \beta\alpha^{n}\left[\left(x_1^{nR} - x_2^{nR}\right) + \left(x_1^{nL} - x_2^{nL}\right) - 2\left(x_1^{nt} - x_2^{nt}\right)\right]\Bigg]$$

$$+\beta\mu^{N}(x_1^{N} - x_2^{N} - x_1^{NL} - x_2^{NL}) + \beta\alpha^{1}(x_1^{1R} - x_2^{1R} - x_1^{1t} - x_2^{1t})$$

$$+\sum_{n=1}^{N}\beta H(x_1^{n} - x_2^{n}) \tag{2.36}$$

where λ_1^n, λ_2^n, μ^n, and α^n are Lagrange multipliers used analogously to the λ in equation (2.10). In the last term, the x_i^n are replaced by equation (2.27).

Taking partial derivatives with respect to the y_{ij} and t_{ij}, because of equation (2.23), $\frac{\partial\Phi}{\partial t_{12}^{n}} = \frac{\partial\Phi}{\partial t_{21}^{n}}$ yields:

$$\frac{\partial\Phi}{\partial y_{11}^{n}} = \beta\left(\varepsilon_{11} + \frac{1}{6}\gamma_{1}^{n} + \frac{1}{6}\gamma_{1}^{n+1}\right) - \frac{5}{6}\ln x_{1}^{n+1}$$

$$-\frac{5}{6}\ln x_{1}^{n} + \ln y_{11}^{n} + \beta\lambda_{2}^{n} + \beta\mu^{n} - \beta\mu^{n+1}$$

$$+\beta\alpha^{n} + \beta\alpha^{n+1} + \frac{2}{6}\beta H \tag{2.37}$$

$$\frac{\partial\Phi}{\partial y_{12}^{n}} = \beta\left(\varepsilon_{12} + \frac{1}{6}\gamma_{1}^{n} + \frac{1}{6}\gamma_{2}^{n+1}\right) - \frac{5}{6}\ln x_{2}^{n+1}$$

$$-\frac{5}{6}\ln x_{1}^{n} + \ln y_{12}^{n} + \beta\lambda_{2}^{n} + \beta\mu^{n} - \beta\mu^{n+1}$$

$$+\beta\alpha^{n} + \beta\alpha^{n+1} \tag{2.38}$$

$$\frac{\partial\Phi}{\partial y_{21}^{n}} = \beta\left(\varepsilon_{21} + \frac{1}{6}\gamma_{1}^{n+1} + \frac{1}{6}\gamma_{2}^{n}\right) - \frac{5}{6}\ln x_{1}^{n+1}$$

$$-\frac{5}{6}\ln x_{2}^{n} + \ln y_{21}^{n} + \beta\lambda_{2}^{n} + \beta\mu^{n} - \beta\mu^{n+1}$$

$$-\beta\alpha^{n} + \beta\alpha^{n+1} \tag{2.39}$$

$$\frac{\partial\Phi}{\partial y_{22}^{n}} = \beta\left(\varepsilon_{22} + \frac{1}{6}\gamma_2^n + \frac{1}{6}\gamma_2^{n+1}\right) - \frac{5}{6}\ln x_2^{n+1}$$

$$-\frac{5}{6}\ln x_2^n + \ln y_{22}^n + \beta\lambda_2^n - \beta\mu^n + \beta\mu^{n+1}$$

$$-\beta\alpha^n - \beta\alpha^{n+1} - \frac{2}{6}\beta H \tag{2.40}$$

$$\frac{\partial\Phi}{\partial t_{11}^{n}} = 2\beta\varepsilon_{11} + \frac{2}{3}\gamma_1^n - \frac{10}{3}\ln x_1^n + 2\ln t_{11}^n + 2\beta\lambda_1^n - \beta\alpha^n + \frac{2}{3}\beta H \tag{2.41}$$

$$\frac{\partial\Phi}{\partial t_{12}^{n}} = 2\beta\varepsilon_{12} + 2\beta\varepsilon_{21} + \frac{2}{3}\beta\gamma_1^n + \frac{2}{3}\beta\gamma_2^n$$

$$-\frac{10}{3}\ln x_1^n - \frac{10}{3}\ln x_2^n + 4\ln t_{12}^n + 4\beta\lambda_1^n - 2\beta\alpha^n + 2\beta\alpha^n \tag{2.42}$$

$$\frac{\partial\Phi}{\partial t_{22}^{n}} = 2\beta\varepsilon_{22} + \frac{2}{3}\beta\gamma_2^n - \frac{10}{3}\ln x_2^n + 2\ln t_{22}^n + 2\beta\lambda_1^n + 2\beta\alpha^n - \frac{2}{3}\beta H \tag{2.43}$$

Setting equations (2.33)-(2.39) equal to zero and solving for the y_{ij}^n and t_{ij}^n yields the extremum values. These values are:

$$y_{11}^n = \left(x_1^n x_1^{n+1}\right)^{5/6} \bullet \exp\left(\beta\left(\varepsilon - \frac{1}{6}\gamma_1^n - \frac{1}{6}\gamma_1^{n+1}\right) - \beta\lambda_2\right)$$

$$\bullet\exp\left(\beta\left(-\mu^n + \mu^{n+1} - \alpha^n - \alpha^{n+1} - \frac{1}{3}H\right)\right) \tag{2.44}$$

$$y_{12}^n = \left(x_1^n x_2^{n+1}\right)^{5/6} \bullet \exp\left(-\beta\left(\varepsilon + \frac{1}{6}\left(\gamma_1^n + \gamma_2^{n+1}\right)\right) - \beta\lambda_2\right)$$

$$\bullet\exp\left(\beta\left(-\mu^n - \mu^{n+1} - \alpha^n + \alpha^{n+1}\right)\right) \tag{2.45}$$

$$y_{21}^n = \left(x_2^n x_1^{n+1}\right)^{5/6} \bullet \exp\left(-\beta\left(\varepsilon + \frac{1}{6}\left(\gamma_1^{n+1} + \gamma_2^n\right)\right) - \beta\lambda_2\right)$$

$$\bullet\exp\left(\beta\left(\mu^n + \mu^{n+1} + \alpha^n - \alpha^{n+1}\right)\right) \tag{2.46}$$

$$y_{22}^{n} = \left(x_2^n x_2^{n+1}\right)^{5/6} \bullet \exp\left(-\beta\left(-\varepsilon + \frac{1}{6}\left(\gamma_2^n + \gamma_2^{n+1}\right)\right) - \beta\lambda_2^n\right)$$

$$\bullet \exp\left(\beta\left(\mu^n - \mu^{n+1} + \alpha^n + \alpha^{n+1} + \frac{1}{3}H\right)\right) \qquad (2.47)$$

$$t_{11}^{n} = \left(x_1^n x_1^n\right)^{5/6} \bullet \exp\left(\beta\left(\varepsilon - \frac{1}{3}\gamma_1^n\right) - \beta\lambda_1^n\right) \bullet \exp\left(\beta\left(\alpha^n - \frac{1}{3}H\right)\right) \qquad (2.48)$$

$$t_{12}^{n} = \left(x_1^n x_2^n\right)^{5/6} \bullet \exp\left(-\beta\left(\varepsilon + \frac{1}{6}\left(\gamma_1^n + \gamma_2^n\right)\right) - \beta\lambda_1^n\right) \qquad (2.49)$$

$$t_{22}^{n} = \left(x_2^n x_2^n\right)^{5/6} \bullet \exp\left(\beta\left(\varepsilon - \frac{1}{3}\gamma_2^n\right) - \beta\lambda_1^n\right) \exp\left(-\beta\left(\alpha^n - \frac{1}{3}H\right)\right) \qquad (2.50)$$

Equations (2.44)-(2.50) can now be solved using the Natural Iteration method.

NATURAL ITERATION METHOD

The Natural Iteration method is a numerical technique designed to solve the equations in the CV method. Without loss of generality, ε can be set equal to one. To solve equations (2.14)-(2.16) for the homogeneous case, the following steps are followed:

1. Choose T/k and let $\beta = 1/kT$.
2. Choose a value for x_1 which fixes $x_2 = 1-x_1$.
3. Calculate $y_{ij}e^{-\beta\lambda}$ using equations (2.14)-(2.16).
4. Using equation (2.1), calculate $e^{\beta\lambda}$.
5. Calculate y_{ij}.
6. Using equations (2.3)-(2.4), calculate x_i.
7. Repeat steps 3-6 until the change in x_i is less than the desired tolerance.

For the non-homogeneous case, the NI method requires more steps. Equations (2.44)-(2.50) can be solved as follows:

1. Choose T/k and set $\beta = 1/kT$.

2. Using the solution for the homogeneous case, set all x_i^n equal to this value for all layers, this is the initial guess.

3. Set all α^n and μ^n equal to zero.

4. Calculate y_{ij} and t_{ij} using equations (2.44)-(2.50).

5. Calculate x_i^{nR}, x_i^{nL}, and x_i^{nt} using equations (2.17)-(2.22).

6. Let $C_1^n = \left(\frac{x_1^{nL}}{x_2^{nL}}\right)\left(\frac{x_2^{nR}}{x_1^{nR}}\right)$ for $n = 2$ to N-1.

   C_1^n is proportional to $e^{4\mu^n}$.

   Let $C_1^N = \left(\frac{x_2^{N}}{x_1^{N}}\right)\left(\frac{x_1^{NL}}{x_2^{NL}}\right)$.

   C_1^N is proportional to $e^{2\mu^N}$.

   Let $C_2^n = \left(\left(\frac{x_1^{nL}}{x_2^{nL}}\right)\left(\frac{x_1^{nR}}{x_2^{nR}}\right)\left(\frac{x_2^{nt}}{x_1^{nt}}\right)\right)^2$ for $n = 2$ to N-1.

   C_2^n is proportional to $e^{-8\alpha^n}$.

   Let $C_2^1 = \left(\frac{x_1^{1R}}{x_2^{1R}}\right)\left(\frac{x_2^{1t}}{x_1^{1t}}\right)$.

   C_2^1 is proportional to $e^{-4\alpha^1}$.

7. Make a guess as to how the μ's and α's should change as follows:

   $$\Delta\mu^n = -\ln\frac{C_1^n}{4} \qquad \Delta\alpha^n = \ln\frac{C_2}{8}$$

   $$\Delta\mu^N = -\ln\frac{C_1^N}{2} \qquad \Delta\alpha^1 = \ln\frac{C_2}{4}$$

8. Calculate $\sum_{n=1}^{N}\left(\left|\Delta\mu^n\right| + \left|\Delta\alpha^n\right|\right)$.

   If this is less than the desired tolerance, then go to step 9; otherwise, go back to step 4.

9. Using equations (2.21) and (2.22), calculate the new x_i^n.

10. Calculate $\sum_{n=1}^{N} \left|\Delta x_i^n\right|$, the sum of the absolute changes in x_i.

    If this is less than the desired tolerance, the solution has been found; otherwise, go back to step 4.

CHAPTER III

RESULTS AND CONCLUSIONS

RESULTS

Figure 3 shows the phase diagrams for the homogeneous system calculated with the subroutine BULK The graph is symmetric about the point $x_1 = x_2 = 0.50$, so only one half is shown. The critical temperature ($T_c = \left(0.5 \bullet \ln\left(\frac{z}{z-2}\right)\right)^{-1}$, where z is the number of nearest neighbors, for a simple cubic lattice $z = 6$ (Bodegom, 1982)) is the lowest temperature at which the mixture is miscible in all proportions. In the figure, T_c is at $x_1 = x_2 = 0.50$.

Figure 4 shows the values of the chemical potential, H, as a function of temperature, T. The different curves represent different starting values of T. For a given starting temperature, the concentration on the coexistence curve is found. This corresponds to a value at zero for H. The temperature is then raised and values of H are calculated which fix the concentration to be the same as at the starting temperature. From the figure, it appears that there is a linear relationship between T and H at a fixed concentration. On closer inspection of the data, however, one sees that this is not true.

As mentioned earlier, the case studied refers to a wall with an attractive interaction (equal to 3) with the A molecules in the nearest layer and a repulsive interaction of the same magnitude with the B molecules. There is no interaction between the wall and all other layers. All the following figures relate to this special case. Figure 5 shows the value of the free energy, F, as a function of the coverage of one type of molecule (say A),

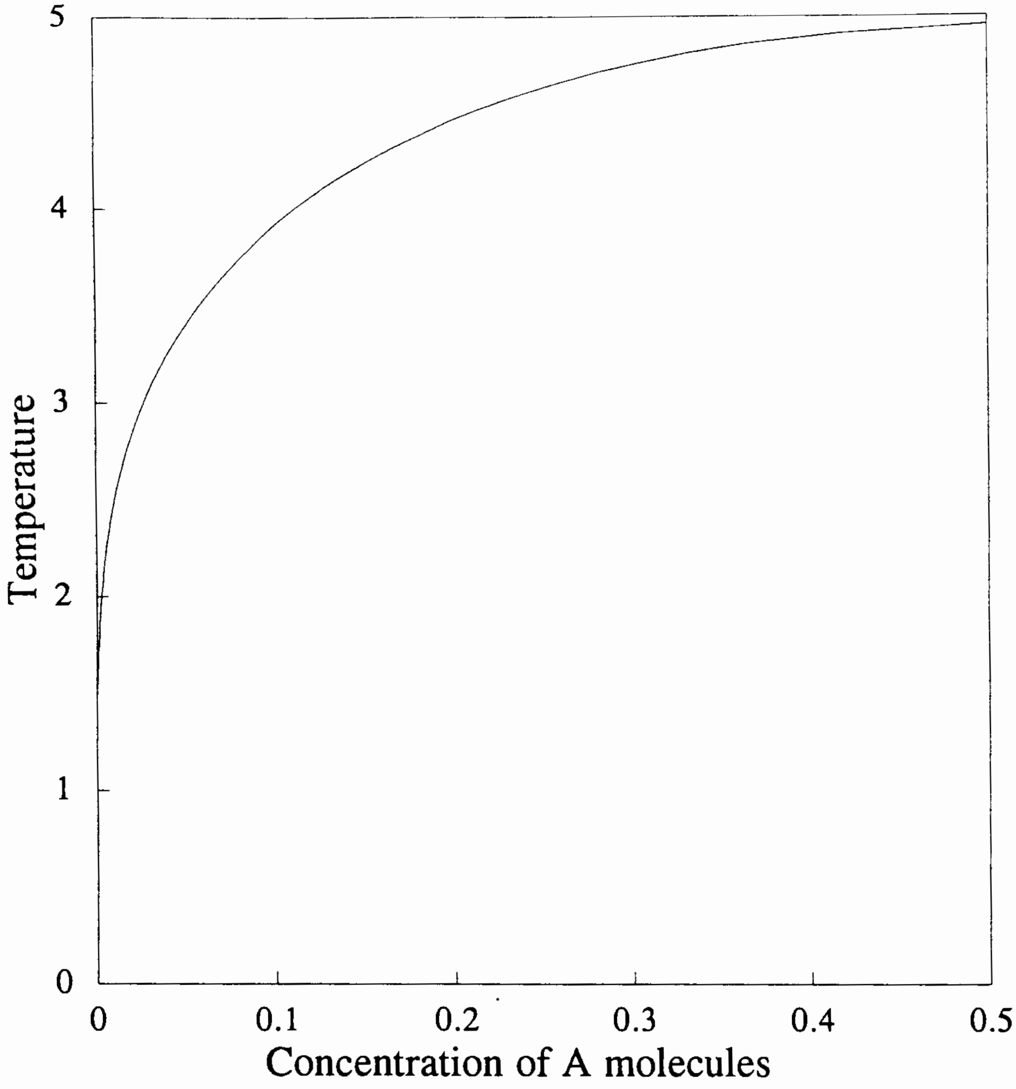

Figure 3. Temperature vs concentration of *A* molecules on the coexistence line.

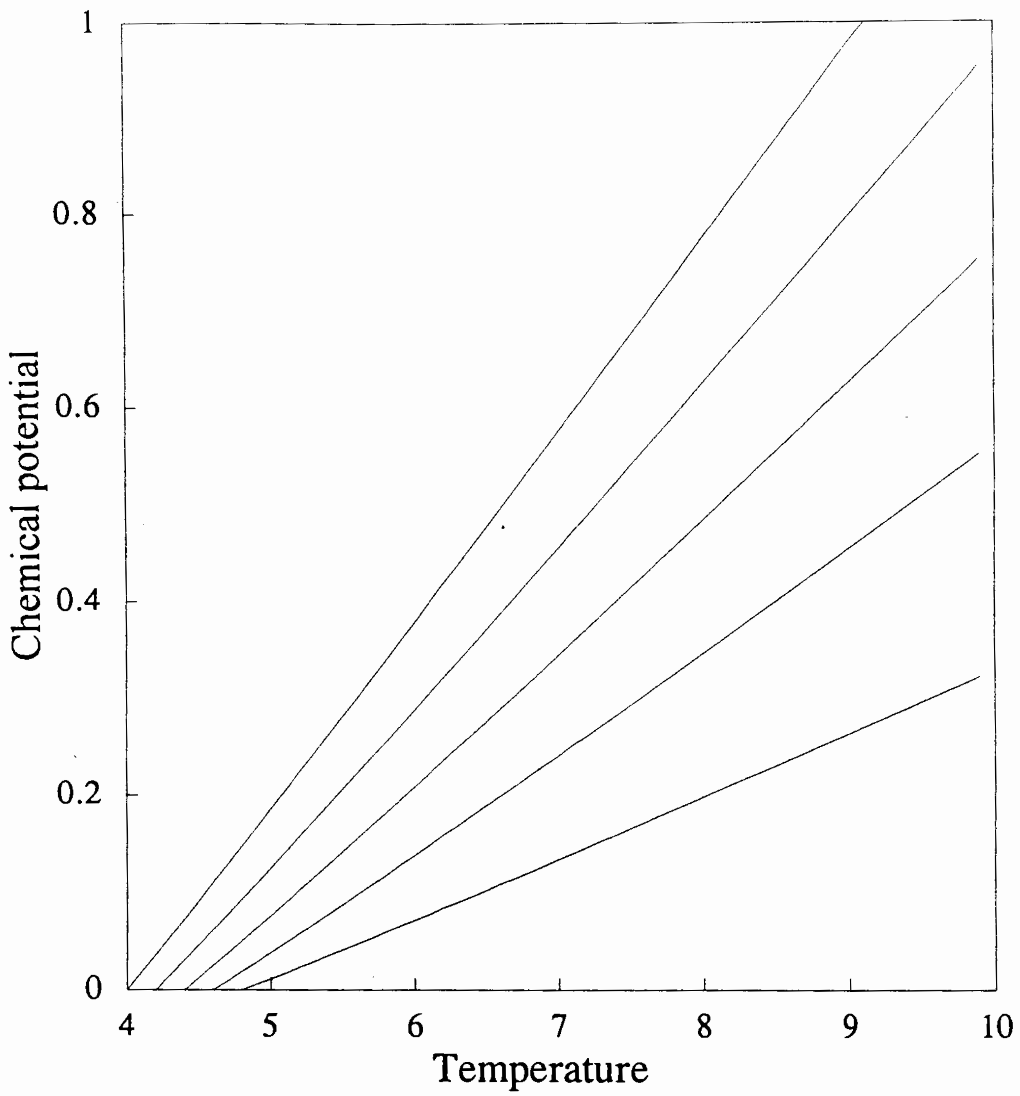

Figure 4. Chemical potential vs temperature for different concentrations.

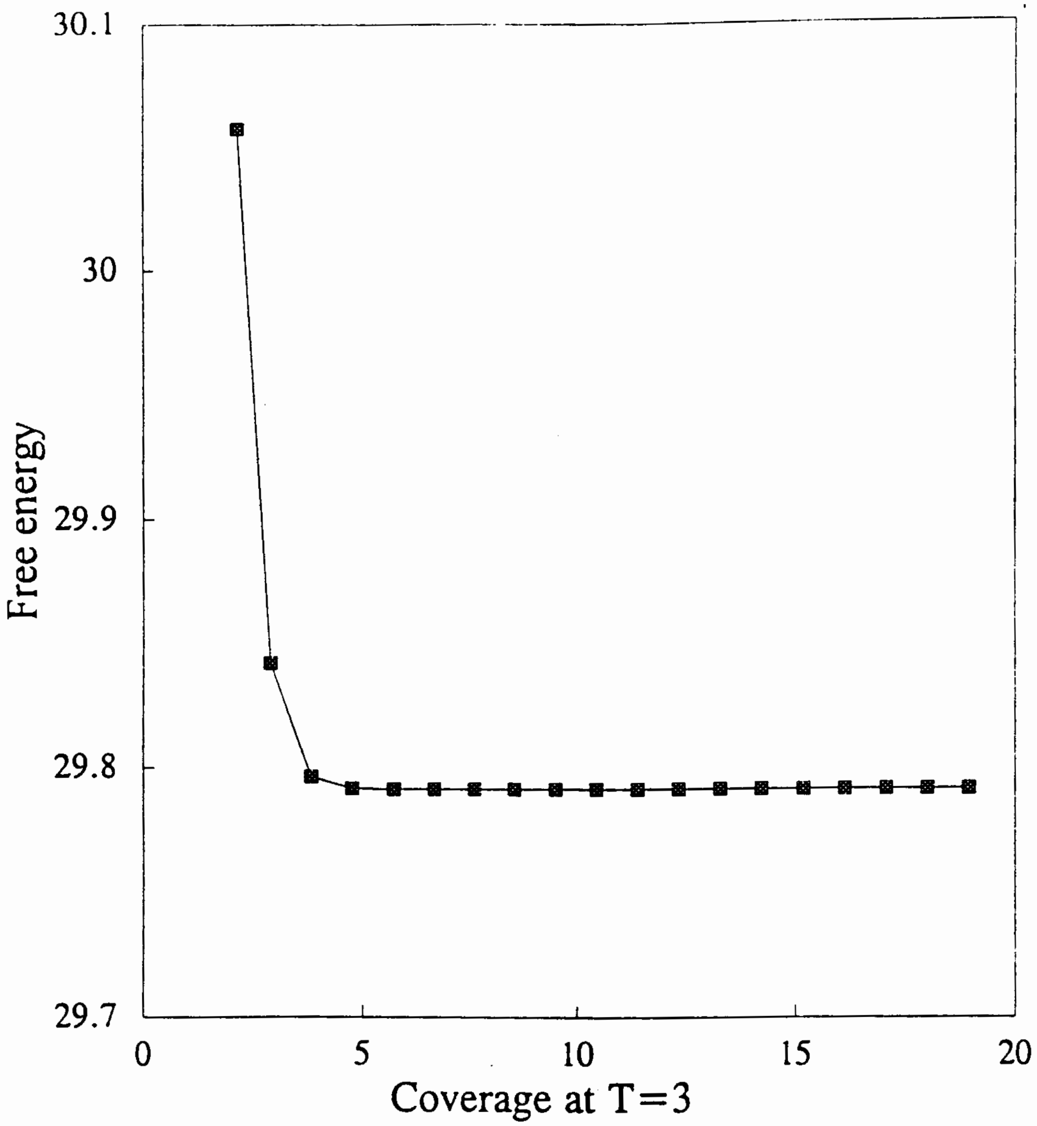

Figure 5. Free energy vs coverage of *A* molecules; the different coverages are obtained by starting with different initial concentration profiles ($T = 3$, the bulk phase is on the coexistence curve).

θ. The coverage is calculated as the total excess over the bulk (near the surface) of one type of molecule. The value of θ thus gives an approximation to the number of layers near the surface where the type of molecule is the opposite of the bulk. The number of layers used in the calculations was 32. Figure 5 was calculated with a temperature of 3 on the coexistence curve. Each point represents a different initial guess for the NI method. The researcher found that the free energy remains the same as the coverage increases because there is no limit on the number of *A* molecules that can migrate from the bulk to the wall. At the boundary away from the wall, the concentration is considered to be that of the bulk system. An improvement to the model would limit the total number of *A* molecules in the system.

Figure 6 shows an enlarged view of Figure 5. One can see that the equilibrium coverage can be any of the values indicated in the figure where the coverage is larger than 7. This is understandable because on the coexistence curve, the free energy is the same if *A* molecules are replaced by *B* molecules and vice versa. The different θ values originate by the translation of the steep part of the gradient in the concentration of the *A* molecules by an integral number of layers (see also Figure 7). Since the concentration of the *A* molecules of one of the bulk phases is almost one, the possible coverages differ by approximately one.

Figure 7 shows a concentration of *A* molecules as a function of the layer number from the surface. The three curves are based on different initial starting values for the NI method. Because the boundary away from the wall is considered to be the value of the bulk system (as discussed previously for Figure 5), the system can keep taking *A* molecules away from the bulk and covering the wall.

Figure 8 shows the free energy, *F*, as a function of coverage, θ, similar to figure 5. This figure, however, is not for the system on the coexistence curve as in Figure 5. Each point again represents a different initial guess for the NI method. This graph should be

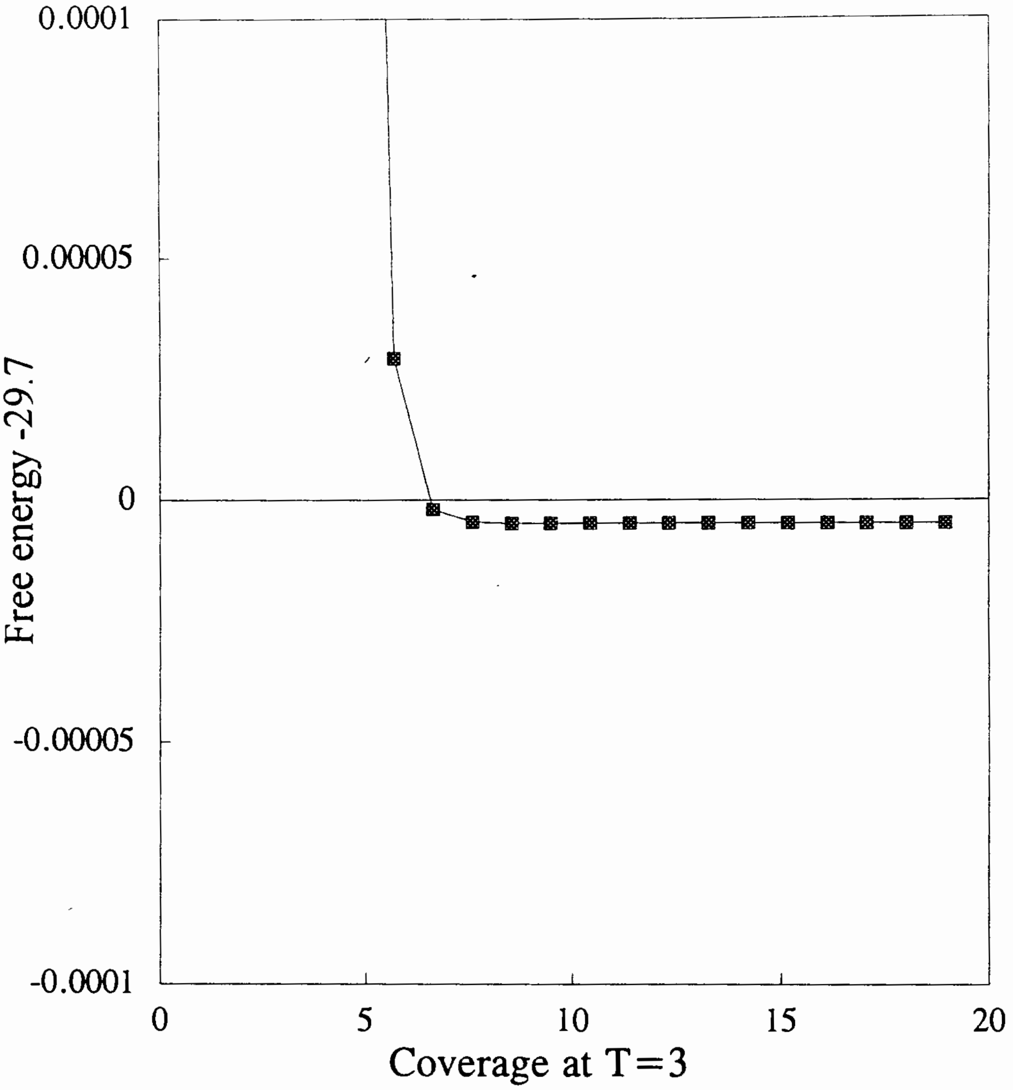

Figure 6. Free energy vs coverage of *A* molecules (magnified from Figure 5).

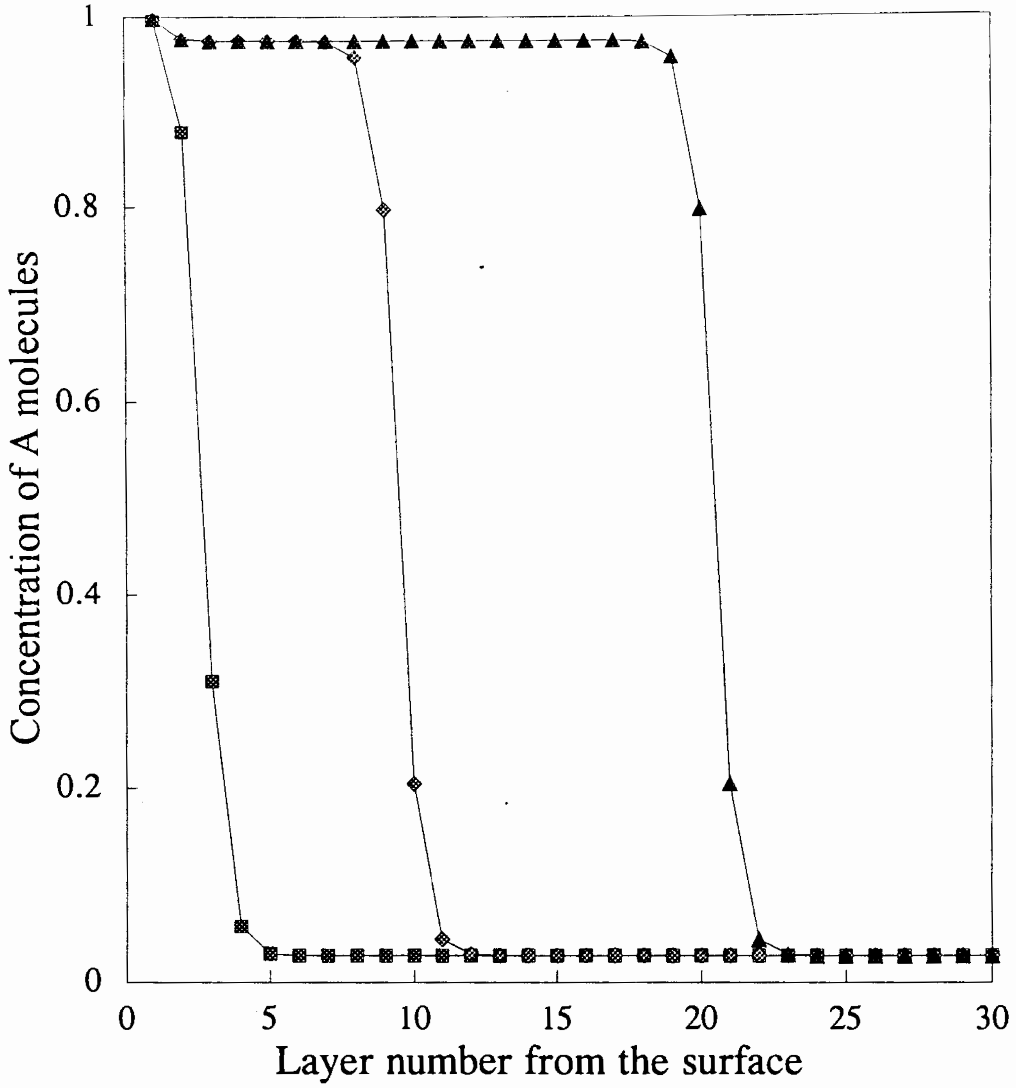

Figure 7. Computed concentration of *A* molecules vs layer number from the surface for different initial input values to the calculations.

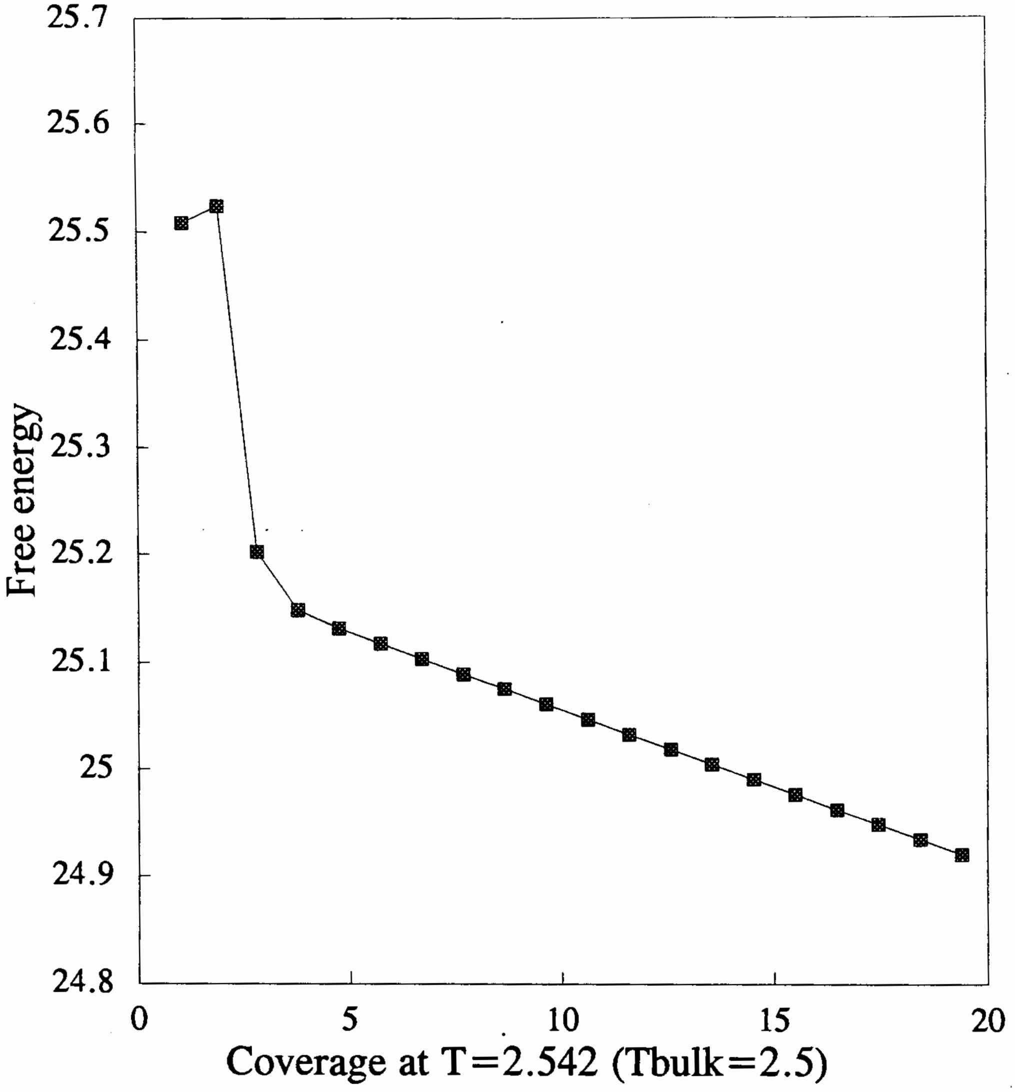

Figure 8. Free energy vs coverage of *A* molecules; the different coverages are obtained by starting with different initial concentration profiles ($T = 2.542$, the bulk phase corresponds to the concentrations of *A* and *B* on the coexistence curve at $T = 2.5$).

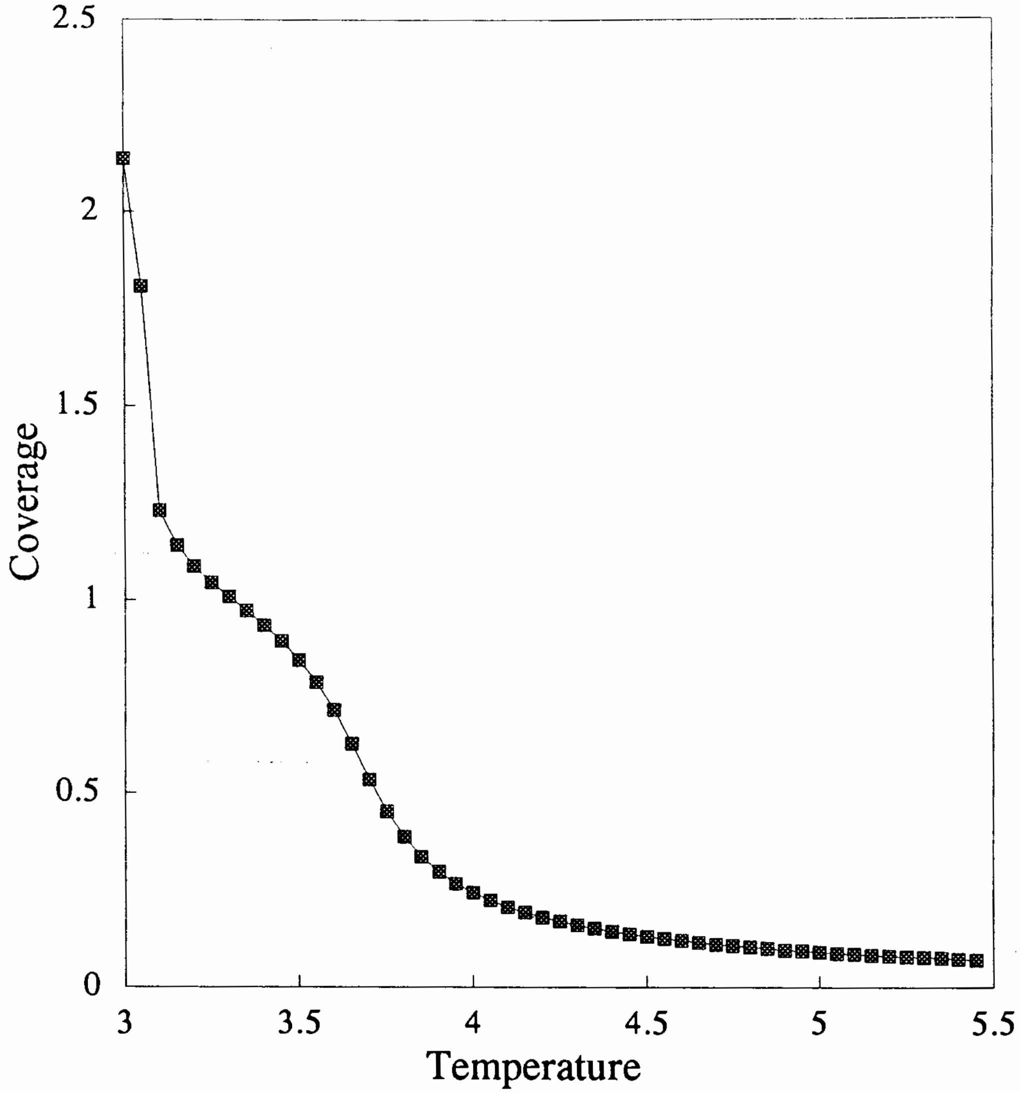

Figure 9. Coverage of *A* molecules vs temperature for the bulk phase corresponding to $T = 3$ on the coexistence curve.

heading for a global minimum for the system and, with more investigation, hopefully this minimum can be found.

Figure 9 shows the coverage for different values of temperature. One should note that the first point (at $T = 3$) is incorrect. At $T = 3$, the coverage can be almost anything (see the discussion regarding Figure 5). One can observe that for lower temperatures, there is a greater coverage of A molecules near the surface than for higher temperatures.

CONCLUSIONS

The DeOliveira and Griffiths model for multilayer absorption on a substrate was extended to the quasichemical or pair approximation using the CV method. The model was modified to deal with a binary mixture. One simple interaction between wall and mixture was considered. This case is similar to the DeOliveira and Griffiths model, i.e., this is a case of a strong substrate, in the terminology of Pandit, et al. (1982). Other substrate-mixture interactions can easily be investigated with the program developed and the results can be compared with the mean field calculations and inferences made by Pandit, et al.

During the latter part of this research, the researcher became aware of a paper by Asada (1990) in which the quasichemical approximation was used to calculate the DeOliveira and Griffiths model. This work showed results similar to those obtained here.

SELECTED REFERENCES

Asada, Hiromer. "Multilayer Lattice Gas Model in the Quasichemical Approximatition," *Surface Science*, *230*, 323-328, 1990.

Beysens, D. and Estere, D. "Adsorption Phenomena at the Surface of Silica Spheres in a Binary Liquid Mixture," *Physical Review Letters*, *54*, 2123-2126, 1985.

Bodegom, Erik. "Super Cooling in Binary Mixtures," Dissertation, Catholic University of America, 1982.

DeOliveira, M.J., and Griffiths, R.B. "Lattice-Gas Model of Multiple Layer Adsorption," *Surface Science*, *71*, 687-694, 1978.

Goldstein, Herbert. *Classical Mechanics*. Addison-Wesley Publishing Company, Inc., p. 46, 1980.

Kikuchi, Ryoichi. "A Theory of Cooperative Phenomena," *Physical Review*, *81(6)*, 988, 1951.

Kikuchi, Ryoichi. "The Cluster Variation Method," *Journal de Physique*, *38*, 67-307, 1977.

Kikuchi, Ryoichi. "Crystal Statistics," (unpublished) Hughes Research Labs, 1979.

Pandit, P., Schick, M., and Wortis, M. "Systematics of Multilayer Adsorption Phenomena on Attractive Substrates," *Physical Review B*, *6(9)*, 5112-5140, 1982.

APPENDIX

COMPUTER PROGRAM

To calculate the concentration profiles using the NI method, a computer program was written in FORTRAN using the Lahey compiler on a personal computer with a 286 processor and a 287 math coprocessor.

The program contains a main program and six subroutines. This appendix contains a listing of the program. The main program initializes the Lagrange multipliers and controls the overall program flow.

Subroutine BULK calculates the x_i for the homogeneous system using the NI method. Subroutine MAGNET calculates the applied magnetic field *H*. This value allows the temperature of the stable system to be raised without changing the values of the x_i. Subroutines LAYER, CALC, CALC2, and CALC3 are used for implementing the NI method for the non-homogeneous system. Subroutine THETA calculates the total coverage of molecules. Subroutine FREE calculates the free energy of the system.

When run on a 486 processor with a 487 math coprocessor, the program executed exponentially quicker.

PROGRAM MAIN

```
      PROGRAM MAIN
*
*     MAIN PROGRAM FOR NATURAL ITERATION METHOD
*
*     A = ALPHA
*     B = BETA
*     E = EPSILON
*     G = GAMMA
*     L = LAMBDA
*     M = MU
*     T = STARTING TEMPERATURE
*     TF = FINAL TEMPERATURE
*     TINY = CONVERSION CRITERIA FOR MU AND ALPHA
*     TINY2 = CONVRESION CRITERIA FOR X'S

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION L1,L2,M
      DIMENSION G1(128),G2(128),L1(128),L2(128)
      CHARACTER C2*3,C3*3,C1*10
      LOGICAL TRACE,TRACE1,TRACE2,TRACE3

      INCLUDE VALUES

      INCLUDE T3

      OPEN(8,FILE='C:TRACE.DAT',FORM='FORMATTED')
*       OPEN(8,FILE='LPT1',STATUS='OLD',FORM='FORMATTED')

      PRINT*,' ENTER THE NUMBER OF LAYERS '
      READ*,NLAYER
      PRINT*,' ENTER THE INITIAL TEMPERATURE '
      READ*,TI
      PRINT*,' ENTER II, WHICH IS 100 * INITIAL TEMPERATURE '
      READ*,II
      PRINT*, 'ENTER ONE POSITIVE VALUE OF GAMMA '
      READ*,GAM

      TRACE = .FALSE.
      TRACE1 = .FALSE.
      TRACE2 = .FALSE.
      TRACE3 = .FALSE.
      C = 1.0/6.0
      D = 1.0/3.0
      E = 1.0
      TINY2 = 1.0E-06 * NLAYER
      TF = 5.5
      MAXIT = 30000

 66   T = TI

      WRITE(C3,FMT='(I3)') II
      C2 = 'C:X'
      C1 = C2//C3//'.DAT'
      OPEN(10,FILE=C1,FORM='FORMATTED')

      C1(3:3) = 'T'
      OPEN(11,FILE=C1,FORM='FORMATTED')

      H = 0.0
      CALL BULK(T,X1A,X2A)
      IF(TRACE)WRITE(8,FMT='('' IN MAIN PROGRAM, X1A = '',E15.5,
     +  '' X2A = '',E15.5)') X1A, X2A

*      DO WHILE T .LT. TF

20      DO 1 I = 1, NLAYER
          A(I) = 0.0
          M(I) = 0.0
          G1(I) = 0.0
          G2(I) = 0.0
*         G1(I) = -2.0/REAL(I)**3
*         G2(I) = G1(I)
          X1(I) = X1A
          X2(I) = X2A
 1      CONTINUE


        G1(1) = -GAM
        G2(1) = GAM

        B = 1.0/T

        DO 2 I = 1, NLAYER-1
          A11(I) = EXP(B*(E - C*G1(I) - C*G1(I+1)))
          A12(I) = EXP(-B*(E + C*G1(I) + C*G2(I+1)))
          A21(I) = EXP(-B*(E + C*G1(I+1) + C*G2(I)))
          A22(I) = EXP(B*(E - C*G2(I) - C*G2(I+1)))
          C11(I) = EXP(B*(E - D*G1(I)))
          C22(I) = EXP(B*(E - D*G2(I)))
          C12(I) = EXP(-B*(E + C*G1(I) + C*G2(I)))
          IF(TRACE)WRITE(8,FMT='('' I = '',I4,'' A11(I) = '',E15.5,
     +      '' A12(I) = '',E15.5/'' A21(I) = '',E15.5,'' A22(I) = '',
     +      E15.5,'' C11(I) = '',E15.5/'' C22(I) = '',E15.5,
     +      '' C12(I) = '',E15.5)')
     +      I,A11(I),A12(I),A21(I),A22(I),C11(I),C22(I),C12(I)

 2     CONTINUE


        C11(NLAYER) = EXP(B*(E - D*G1(NLAYER)))
        C22(NLAYER) = EXP(B*(E - D*G2(NLAYER)))
        C12(NLAYER) = EXP(-B*(E + C*G1(NLAYER) + C*G2(NLAYER)))
        IF(TRACE)WRITE(8,FMT='('' I = '',I4,'' C11(I) = '',E15.5,
     +      '' C22(I) = '',E15.5/'' C12(I) = '',E15.5)')
     +    NLAYER,C11(NLAYER),C22(NLAYER),C12(NLAYER)

        MOVES = 0
        TINY = .01*NLAYER
        ITER = 1

 10     CALL LAYER(TINY)
        SUM = 0.0
        DO 5 I = 1, NLAYER
          SUM = SUM + ABS(X1(I) - X1T(I)/(X1T(I) + X2T(I)))
  5     CONTINUE
        IF(TRACE3)WRITE(8,FMT='('' **** IN MAIN ** SUM = '',E15.5)') SUM


        IF(SUM.GT.TINY2) THEN
          DO 30 I = 1, NLAYER
            S = X1T(I) + X2T(I)
            X1(I) = X1T(I)/S
            X2(I) = X2T(I)/S
            IF(TRACE)WRITE(8,FMT='('' I = '',I4,'' X1(I) = '',E15.5,
     +        '' X2(I) = '',E15.5)') I, X1(I), X2(I)

 30       CONTINUE
          TINY = SUM/5.0
          ITER = ITER + 1
          IF(ITER .GT. MAXIT) THEN
            WRITE(8,FMT='('' ITERATIONS EXCEEDED IN MAIN  SUM = '',
     +      E15.5,'' II = '',I4)') SUM,II
            DO 32 I = 1,NLAYER
              WRITE(8,FMT='(E15.5)') X1(I)
 32         CONTINUE
            GO TO 70
          ENDIF
          GO TO 10
        ELSE
*         *--- HAVE CONVERGED X'S
          IF(TRACE1) WRITE(8,FMT='('' IN MAIN, ITER = '',I6,
     +               '' SUM = '',E14.5)') ITER, SUM

          DO 40 I = 1,NLAYER
            WRITE(10,FMT='(4(E15.5,5X))') X1(I),T,A(I),M(I)
40       CONTINUE
         WRITE(10,*)

*        *--- CALCULATE COVERAGE
         CALL THETA(X1(1),X1A,NLAYER,SUMTH)

*        *--- CALCULATE FREE ENERGY
         CALL FREE(G1,G2,FREEEN)

         WRITE(11,FMT='(3E20.10)') SUMTH,T,FREEEN

         IF(MOVES .GT. 0)THEN
           DELS = ABS(SUMTH-SUMTHO)
           IF(DELS .GT. 1.0E-2 .AND. MOVES .LT. 20 ) THEN
*            *--- MOVE LAYERS
             DO 55 I = NLAYER-2, 1
               X1(I+1) = X1(I)
               X2(I+1) = X2(I)
  55         CONTINUE
             MOVES = MOVES + 1
             ITER = 1
             TINY = .001 * NLAYER
             SUMTHO = SUMTH
             GO TO 10
           ELSEIF(DELS .GT. 1.0E-2)THEN
               WRITE(10,FMT='('' DID NOT CONVERGE AFTER MAX MOVES'')')
               WRITE(11,FMT='('' DID NOT CONVERGE AFTER MAX MOVES'')')
           ENDIF
         ELSE
*          *--- MOVE LAYERS
           DO 56 I = NLAYER - 2, 1
             X1(I+1) = X1(I)
             X2(I+1) = X2(I)
  56       CONTINUE
           MOVES = MOVES + 1
           ITER = 1
           TINY = .001 * NLAYER
           SUMTHO = SUMTH
           GO TO 10
         ENDIF

        ENDIF

 70     T = T + .05

        IF(T .LE. TF) THEN
          CALL MAGNET(T,X1A,H)
          IF(TRACE1)WRITE(8,FMT='('' AFTER MAGNET CALL, H = '',E15.5)')H
          GO TO 20
        ENDIF
*     *--- END WHILE T .LT. TF

*     *--- START AT NEW TEMPERATURE

75    II = II + 25
      TI = TI + .25
      IF(TI.LT.TF) GO TO 66

      END
```

SUBROUTINE BULK

```
      SUBROUTINE BULK(KT,X1,X2)
*
*     NATURAL ITERATION METHOD FOR GIVEN TEMPERATURE FOR BULK MIXTURE
*

*     KT - INPUT - TEMPERATURE
*     X1 - OUTPUT - CONCENTRATION OF X1
*     X2 - OUTPUT - CONCENTRATION OF X2


      LOGICAL  TRACE, TRACE1
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION KT

*     Z1 = EXP(-BETA*LAMDA)*Y1  SAME FOR Z2 AND Z3
*     Z4 = EXP(-BETA * LAMDA)

      TRACE = .FALSE.
      TRACE1 = .FALSE.
      TINY = 1.0D-13
      X1 = .25
      X2 = 1.0 - X1
      X1OLD = X1
      E1 = 5.0/6.0
      EPS = 1.0
      IF(TRACE1) WRITE(8,90) KT
90    FORMAT(' ENTERING SUBROUTINE BULK, KT = ',E15.5)

      ITER = 1
      BETA = 1.0/KT
      E2 = EXP(BETA*EPS)
      E3 = EXP(-BETA*EPS)

*     *---DO WHILE X1-X1OLD .GT. TINY

  10      IF(TRACE) WRITE(8,110) ITER
 110      FORMAT (' ******',' ITERATION = ',I5,1X,'*******')

          Z1 = (X1*X1)**E1 * E2
          Z2 = (X1*X2)**E1 * E3
          Z3 = (X2*X2)**E1 * E2
          Z4 = Z1 + 2.0*Z2 + Z3
          IF(TRACE) WRITE(8,115) Z1, Z2, Z3, Z4
 115      FORMAT (' Z1 = ',F15.7,'  Z2 = ',F15.7,
     +            '  Z3 = ',F15.7,' Z4 = ',F15.7)

          X1 = (Z1 + Z2)/Z4
          X2 = (Z2 + Z3)/Z4
          IF(TRACE) WRITE(8,125) X1, ABS(X1-X1OLD)
 125      FORMAT (' X1 = ',E15.7,' X1-X1OLD = ',E15.7//)

*         *--- CHECK EVERY 10 ITERATIONS
          IF(MOD(ITER,10) .EQ. 0) THEN
            IF(ABS(X1 - X1OLD) .LT. TINY) THEN
*             *-- FOUND SOLUTION
              GO TO 999
            ELSE
              X1OLD = X1
              IF(TRACE) WRITE(8,130) X1OLD
 130          FORMAT (' X1OLD = ',E15.7)
              ITER = ITER + 1
              GO TO 10
            ENDIF
          ELSE
            ITER = ITER + 1
            GO TO 10
          ENDIF

*     *--- END WHILE

999   X2 = 1.0D0 - X1
      IF(TRACE1) WRITE(8,250) X1,X2,ITER
250   FORMAT(' LEAVING SUBROUTINE BULK'/' X1 = ',E15.7,' X2 = ',E15.7,
     +       ' ITER = ',I10)

      END
```

SUBROUTINE CALC

```
      SUBROUTINE CALC(I)
*
*     CALCLULATE X'S
*

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION M
      LOGICAL TRACE,TRACE1,TRACE2

      INCLUDE VALUES

      B11 = EXP(B*(-M(I) + M(I+1) - A(I) - A(I+1)))
      B12 = EXP(B*(-M(I) - M(I+1) - A(I) + A(I+1)))

      B21 = EXP(B*(M(I) + M(I+1) + A(I) - A(I+1)))
      B22 = EXP(B*(M(I) - M(I+1) + A(I) + A(I+1)))

      D11 = EXP(B*A(I))

*     D22 = EXP(-B*A(I))
*     *--- D22 = 1/D11

      Y11(I) = Y11P(I) * B11
      Y12(I) = Y12P(I) * B12
      Y21(I) = Y21P(I) * B21
      Y22(I) = Y22P(I) * B22
      T11(I) = T11P(I) * D11
      T22(I) = T22P(I) * 1.0/D11
      IF(TRACE) WRITE(8,FMT='('' '',30(''*''),'' IN CALC '',30(''*'')/
     +   '' I = '',I4,'' B11 = '',E15.5,
     +   '' B12 = '',E15.5/'' B21 = '',E15.5,'' B22 = '',E15.5,
     +   '' D11 = '',E15.5)') I, B11, B12, B21, B22, D11

      X1R(I) = Y11(I) + Y12(I)
      X2R(I) = Y21(I) + Y22(I)
      IF(TRACE) WRITE(8,FMT='('' X1R = '',E15.5,'' X2R = '',E15.5)')
     +   X1R(I),X2R(I)

      IF(I.NE.1) THEN
        X1L(I) = Y11(I-1) + Y21(I-1)
        X2L(I) = Y12(I-1) + Y22(I-1)
        IF(TRACE) WRITE(8,FMT='('' X1L = '',E15.5,'' X2L = '',E15.5)')
     +    X1L(I),X2L(I)

      ENDIF

      X1T(I) = T11(I) + T12(I)
      X2T(I) = T12(I) + T22(I)
      IF(TRACE) WRITE(8,FMT='('' X1T = '',E15.5,'' X2T = '',E15.5/
     +    '' '',30(''*''),'' LEAVING CALC '',30(''*''))')
     +  X1T(I),X2T(I)

      END
```

SUBROUTINE CALC2

```
      SUBROUTINE CALC2(I)
*
*     CALCLULATE YP'S AND TP'S
*

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION M
      LOGICAL TRACE,TRACE1,TRACE2

      INCLUDE VALUES

      F = EXP(2.0*B*H)
      E = 5.0/6.0

      Y11P(I) = (X1(I)*X1(I+1)*F)**E * A11(I)
      Y12P(I) = (X1(I)*X2(I+1))**E * A12(I)
      Y21P(I) = (X2(I)*X1(I+1))**E * A21(I)
      Y22P(I) = (X2(I)*X2(I+1)*1.0/F)**E * A22(I)
      T11P(I) = (X1(I)*X1(I)*F)**E * C11(I)
      T22P(I) = (X2(I)*X2(I)*1.0/F)**E * C22(I)
      T12(I) = (X1(I)*X2(I))**E * C12(I)

      IF(TRACE1) WRITE(8,FMT='('' '',30(''*''),''IN CALC2'',30(''*'')/
     +  '' B = '',E15.5,'' H = '',E15.5,'' F = '',E15.5/
     +  '' I = '',I4,'' Y11P = '',E15.5,'' Y12P = '',E15.5/
     +  '' Y21P = '',E15.5,'' Y22P = '',E15.5,'' T11P = '',E15.5/
     +  '' T22P = '',E15.5,'' T12 = '',E15.5/
     +  '' '',30(''*''),'' LEAVING CALC2 '',30(''*''))')
     +  B,H,F,I,Y11P(I),Y12P(I),Y21P(I),Y22P(I),T11P(I),T22P(I),T12(I)

      END
```

SUBROUTINE CALC3

```
      SUBROUTINE CALC3(I)
*
*     CALCLULATE X'S AT NLAYER
*

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION M
      LOGICAL TRACE,TRACE1,TRACE2

      INCLUDE VALUES

      F = EXP(2.0*B*H)
      E = 5.0/6.0

      T11P(I) = (X1(I)*X1(I)*F)**E * C11(I)
      T22P(I) = (X2(I)*X2(I)*1.0/F)**E * C22(I)
      T12(I) = (X1(I)*X2(I))**E * C12(I)
      D11 = EXP(B*A(I))
      T11(I) = T11P(I) * D11
      T22(I) = T22P(I) * (1.0/D11)

      X1T(I) = T11(I) + T12(I)
      X2T(I) = T12(I) + T22(I)
      X1L(I) = Y11(I-1) + Y21(I-1)
      X2L(I) = Y12(I-1) + Y22(I-1)

      IF(TRACE) WRITE(8,FMT='('' '',30(''*''),''IN CALC3'',30(''*'')/
     +  '' B = '',E15.5,'' H = '',E15.5,'' F = '',E15.5/
     +  '' I = '',I4,'' T11P = '',E15.5,'' T22P = '',E15.5/
     +  '' T12 = '',E15.5,'' D11 = '',E15.5,'' T11 = '',E15.5/
     +  '' T22 = '',E15.5,'' X1T = '',E15.5/'' X2T = '',E15.5,
     +  '' X1L = '',E15.5,'' X2L = '',E15.5/
     +  '' '',30(''*''),'' LEAVING CALC3 '',30(''*''))')
     +  B,H,F,I,T11P(I),T22P(I),T12(I),D11,T11(I),T22(I),X1T(I),
     +  X2T(I),X1L(I),X2L(I)

      END
```

SUBROUTINE FREE

```
      SUBROUTINE FREE(G1,G2,FREEEN)
*
*     CALCULATE FREE ENERGY
*
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION M
      DIMENSION G1(128),G2(128)
      LOGICAL TRACE,TRACE1,TRACE2

      INCLUDE VALUES

      A1 = 1.0/6.0
      B1 = 2.0/3.0
      E = 1.0

      SUM1 = 0.0
      SUM2 = 0.0
      SUM3 = 0.0
      SUM4 = 0.0
      SUM5 = 0.0
      SUM6 = 0.0
      SUM7 = 0.0
      SUM8 = 0.0

      SUM8 = SUM8 + X1(1) - X2(1)

      DO 1 I = 2, NLAYER - 1

        SUMY1 = Y11(I) + Y12(I) + Y21(I) + Y22(I)
        SUMY2 = Y11(I-1) + Y12(I-1) + Y21(I-1) + Y22(I-1)

        SUMT = T11(I) + 2.0*T12(I) + T22(I)

        SUM1 = SUM1 + ( -E*T11(I) + 2.0*E*T12(I) - E*T22(I))/SUMT
        SUM2 = SUM2 + ( - E*Y11(I-1) + E*Y12(I-1) + E*Y21(I-1)
     1                  - E*Y22(I-1))/SUMY2

        SUM3 = SUM3 +  G1(I)*(A1*(Y11(I)/SUMY1 + Y11(I-1)/SUMY2)
     1                  + B1*T11(I)/SUMT
     1              + A1*(Y12(I)/SUMY1 + Y21(I-1)/SUMY2)
     1              + B1*T12(I)/SUMT)

        SUM4 = SUM4 + G2(I)*(A1*(Y21(I)/SUMY1 + Y12(I-1)/SUMY2)
     1                  + B1*T21(I)/SUMT
     1              + A1*(Y22(I)/SUMY1 + Y22(I-1)/SUMY2)
     1              +  B1*T22(I)/SUMT)

        SUM5 = SUM5 + X1(I)*(LOG(X1(I)) - 1.0) + X2(I)*(LOG(X2(I)) - 1.0)

        SUM6 = SUM6 + T11(I)/SUMT*(LOG(T11(I)/SUMT) - 1.0)
     1         +  2.0*T12(I)/SUMT*(LOG(T12(I)/SUMT) - 1.0)
     1         +      T22(I)/SUMT*(LOG(T22(I)/SUMT) - 1.0)


        SUM7 = SUM7 + Y11(I)/SUMY1*(LOG(Y11(I)/SUMY1) - 1.0)
     1         +      Y12(I)/SUMY1*(LOG(Y12(I)/SUMY1) - 1.0)
     1         +      Y21(I)/SUMY1*(LOG(Y21(I)/SUMY1) - 1.0)
     1         +      Y22(I)/SUMY1*(LOG(Y22(I)/SUMY1) - 1.0)
```

```
       SUM8 = SUM8 + X1(I) - X2(I)
       IF(TRACE) WRITE(8,FMT='('' SUMY1 = '',E15.5/'' SUMY2 = '',
    +    E15.5/'' SUMT = '',E15.5/'' SUM1 = '',E15.5/'' SUM2 = '',
    +    E15.5/'' SUM3 = '',E15.5/'' SUM4 = '',E15.5/'' SUM5 = '',
    +    E15.5/'' SUM6 = '',E15.5/'' SUM7 = '',E15.5/'' SUM8 = '',
    +    E15.5)') SUMY1, SUMY2, SUMT, SUM1, SUM2, SUM3, SUM4, SUM5,
    +    SUM6, SUM7, SUM8

1    CONTINUE


      SUM8 = SUM8 + X1(NLAYER) - X2(NLAYER)


      FREEEN = B*(2.0*SUM1 + SUM2 + SUM3 + SUM4)
   1        - 5.0*SUM5 + 2.0*SUM6 + SUM7 + B*H*SUM8
      IF(TRACE) WRITE(8,FMT='('' FREEEN = '',E15.5)') FREEEN

      END
```

SUBROUTINE LAYER

```
      SUBROUTINE LAYER(TINY)
*
*     CALCLULATE LAGRANGE MULTIPLIERS MU AND ALPHA
*
*     A = ALPHA
*     B = BETA
*     E = EPSILON
*     G = GAMMA
*     L = LAMBDA
*     M = MU
*     MAXIT = MAXIMUM NUMBER OF ITERATIONS ALLOWED
*     ITER = NUMBER OF ITERATIONS PERFORMED

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DOUBLE PRECISION M
      LOGICAL TRACE,TRACE1,TRACE2,TRACE3

      INCLUDE VALUES

      INCLUDE T3

      MAXIT = 500
      ITER = 1

      DO 2 I = 1, NLAYER-1
        CALL CALC2(I)
2     CONTINUE

*     *--- CALCULATE FIRST LAYER
5     CALL CALC(1)
*     *--- C1 DOES NOT EXIST
      C2 = (X1R(1)/X2R(1)) * (X2T(1)/X1T(1))
      DELA = LOG(C2)/4.0
      A(1) = A(1) + DELA
      SUM = ABS(DELA)
      IF(TRACE) WRITE(8,FMT='('' '',30(''*''),'' IN LAYER '',30(''*'')/
     +  '' TINY = '',E15.5/
     +  '' C2 = '',E15.5,'' DELA = '',E15.5,'' A(1) = '',E15.5/
     +  '' M(1) = '',E15.5,'' SUM = '',E15.5)')
     +  TINY,C2,DELA,A(1),M(1),SUM
      IF(TRACE3) WRITE(8,FMT='('' I = 1, DELA = '',E15.5)') DELA

*    *--- CALCULATE LAYER 2 THRU M-1

      DO 10 I = 2, NLAYER - 1

        CALL CALC(I)

        C1 = (X1L(I)/X2L(I)) * (X2R(I)/X1R(I))
        DELM = -LOG(C1)/4.0

        C2 = (X1L(I)/X1T(I)) * (X1R(I)/X1T(I)) * (X2T(I)/X2L(I)) *
     +       (X2T(I)/X2R(I))
        DELA = LOG(C2)/8.0

        SUM = SUM + ABS(DELA) + ABS(DELM)
        A(I) = A(I) + DELA
        M(I) = M(I) + DELM
        IF(TRACE)WRITE(8,FMT='('' I = '',I4,'' C1 = '',E15.5,'' C2 = ''
     +    ,E15.5/'' DELM = '',E15.5,'' DELA = '',E15.5,'' SUM = '',
     +     E15.5/'' A(I) = '',E15.5,'' M(I) = '',E15.5)')
     +     I,C1,C2,DELM,DELA,SUM,A(I),M(I)
        IF(TRACE3)WRITE(8,FMT='('' I = '',I4,'' DELA = '',E15.5,
     +   '' DELM = '',E15.5)') I, DELA, DELM

 10   CONTINUE

*     *--- CALCULATE MTH LAYER

*     *--- C2 DOES NOT EXIST
      CALL CALC3(NLAYER)
      C1 = (X2A/X1A) * (X1L(NLAYER)/X2L(NLAYER))
      DELM = -LOG(C1)/2.0
      SUM = SUM + ABS(DELM)
      M(NLAYER) = M(NLAYER) + DELM
      IF(TRACE)WRITE(8,FMT='(
    +  '' C1 = '',E15.5/'' DELM = '',E15.5,
    +  '' SUM = '',E15.5,'' M(NALYER) = '',E15.5)')
    +  C1,DELM,SUM,M(NLAYER)
      IF(TRACE3)WRITE(8,FMT='('' I = '',I4,'' DELM = '',E15.5)') NLAYER,
    + DELM

      IF(SUM .GT. TINY) THEN
*       *--- TRY AGAIN
        ITER = ITER + 1
        IF(ITER.GT.MAXIT) THEN
          WRITE(8,FMT='('' ITERATIONS IN LAYER EXCEEDED'')')
          STOP
        ENDIF
        GO TO 5
      ELSE
*       FOUND ALPHA AND MU
        IF(TRACE1)
    +  WRITE(8,FMT='('' IN LAYER, ITER = '',I6,'' SUM = '',E14.5)')
    +  ITER, SUM
        RETURN
      ENDIF

      END
```

SUBROUTINE MAGNET

```
      SUBROUTINE MAGNET(KT,X1,H)
*
*    CALCULATE MAGNETIC FIELD H FOR A GIVEN CONCENTRATION AND TEMPERATURE
*

*     KT - INPUT - TEMPERATURE
*     X1 - INPUT - CONCENTRATION OF X1
*     H  - INPUT/OUTPUT - MAGNETIC FIELD

      LOGICAL  TRACE, TRACE1
      IMPLICIT DOUBLE PRECISION(A-H,O-Z)
      DOUBLE PRECISION KT

*     Z1 = EXP(-BETA*LAMDA)*Y1  SAME FOR Z2 AND Z3
*     Z4 = EXP(-BETA * LAMDA)


      TRACE = .FALSE.
      TRACE1 = .FALSE.

      TINY = 1.0D-10
      X2 = 1.0 - X1
      R = X1/X2
      HOLD = H
      E1 = 5.0/6.0
      EPS = 1.0
      IF(TRACE1) WRITE(8,90) KT,X1
90    FORMAT(' ENTERING SUBROUTINE MAGNET, KT = ',E15.5,'  X1 = ',E15.5)

      ITER = 1
      BETA = 1.0/KT
      E2 = EXP(BETA*EPS)
      E3 = EXP(-BETA*EPS)

*     *---DO WHILE X1-X1OLD .GT. TINY

 10       IF(TRACE) WRITE(8,110) ITER
 110      FORMAT (' ******',' ITERATION = ',I5,1X,'*******')

          E4 = EXP(2.0*BETA*H)
          E5 = EXP(-2.0*BETA*H)
          Z1 = (X1*X1*E4)**E1 * E2
          Z2 = (X1*X2)**E1 * E3
          Z3 = (X2*X2*E5)**E1 * E2
          Z4 = Z1 + 2.0*Z2 + Z3
          IF(TRACE) WRITE(8,115) Z1, Z2, Z3, Z4
 115      FORMAT (' Z1 = ',F15.7,'  Z2 = ',F15.7,
     +             '  Z3 = ',F15.7,' Z4 = ',F15.7)

          X1 = (Z1 + Z2)/Z4
          X2 = (Z2 + Z3)/Z4
          S = (X1/X2)*(1.0/R)
          H = HOLD - (1.0/BETA)*0.60*LOG(S)
          IF(TRACE) WRITE(8,125) H, ABS(H-HOLD)
 125      FORMAT (' H = ',E15.7,' H-HOLD = ',E15.7//)

*         *--- CHECK EVERY 10 ITERATIONS
          IF(MOD(ITER,10) .EQ. 0) THEN
            IF(ABS(H - HOLD) .LT. TINY) THEN
```

```
*             *-- FOUND SOLUTION
              GO TO 999
            ELSE
              HOLD = H
              IF(TRACE) WRITE(8,130) HOLD
 130          FORMAT (' HOLD = ',E15.7)
              ITER = ITER + 1
              GO TO 10
            ENDIF
          ELSE
            ITER = ITER + 1
            GO TO 10
          ENDIF

*     *--- END WHILE

999   IF(TRACE1) WRITE(8,250) H,ITER
250   FORMAT(' LEAVING SUBROUTINE MAGNET'/' H = ',E15.7,
     +       ' ITER = ',I10)

      END
```

SUBROUTINE THETA

```
      SUBROUTINE THETA(X1,X1A,NLAYER,SUMTH)

*       CALCULATE PERCENT COVERAGE

      DOUBLE PRECISION X1(NLAYER),X1A,SUMTH

      SUMTH = 0.0
       DO 1 I = 1, NLAYER
        SUMTH = SUMTH + X1(I)
1     CONTINUE

      SUMTH = SUMTH - REAL(NLAYER)*X1A

       END
```