Abstract

HITCHCOCK, MONICA RENEE. Monte Carlo simulations of complete phase equilibria for binary mixtures. (Under the direction of Carol K. Hall.)

The objective of this thesis is to study the phase equilibria of binary mixtures using molecular simulation. Vapor-liquid, vapor-solid, liquid-liquid, and liquid-solid coexistence lines are calculated for binary mixtures of Lennard-Jones spheres using Monte Carlo simulation and the Gibbs-Duhem integration technique. Complete phase diagrams, *i.e.*, showing all types equilibrium between vapor, liquid, and solid phases are constructed. The calculations presented in this thesis mark the first time that molecular simulation has been used to obtain phase diagrams describing all types of equilibria between vapor, liquid, and solid phases. We present complete phase diagrams for binary Lennard-Jones mixtures with diameter ratios ranging from 0.85 to 0.95 and attractive well-depth ratios ranging from 0.45 to 1.6, at reduced pressures ranging from 0.002 to 0.1. The Lorentz-Berthelot combining rules are used to calculate the cross-species interaction parameters. We systematically explore how the complete phase diagrams change as a function of the diameter ratio, well-depth ratio, binary interaction parameter, and system pressure.

We first calculate complete phase diagrams for several binary mixtures at a single pressure and find that for well-depth ratios of unity (equal attractions among species) there is no interference between the vapor-liquid and solid-liquid coexistence regions. As the well-depth ratio increases or decreases from unity, the vapor-liquid and solid-liquid phase envelopes widen and interfere with each other, leading

to the appearance of a solid-vapor coexistence region. For diameter ratios of 0.95, the solid-liquid lines have a shape characteristic of a solid solution (with or without a minimum melting temperature); as the diameter ratio decreases the solid-liquid lines fall to lower temperatures until they eventually drop below the solid-solid coexistence region, resulting in either a eutectic or peritectic three-phase line.

We then vary the binary interaction parameter in the Berthelot combining rule to study the effect of unlike pair attractions on binary mixture phase behavior. When the binary interaction parameter is unity we find a vapor-liquid coexistence region with a eutectic solid-liquid coexistence region. These two regions are separated by a completely miscible liquid phase. When the binary interaction parameter is less than unity we find that the vapor-liquid and solid-liquid coexistence regions interfere. This interference results in the appearance of a vapor-solid coexistence region bounded above and below by solid-liquid-vapor coexistence lines. We also find that when the binary interaction parameter is less than unity, there is a region of liquid-liquid immiscibility that is metastable with respect to the solid-fluid phase equilibria.

Next we calcuate temperature versus composition phase diagrams for one mixture at five reduced pressures in order to examine the effects of pressure on complete phase behavior. We observe interference between the vapor-liquid and solid-liquid coexistence regions at the lowest pressure. As the pressure increases, the vapor-liquid coexistence region shifts to higher temperatures, while the solid-liquid coexistence region remains essentially unaffected. Eventually, the vapor-liquid coex-

istence region lifts off the solid-liquid coexistence region, ending the interference. We then present pressure versus temperature projections for several mixtures to explore how the three-phase loci change with variations in diameter ratio and well-depth ratio. We find that as the diameter ratio decreases, the maximum pressure in the solid-liquid-vapor locus decreases and the characteristic shape of the solid-liquid coexistence region changes from peritectic to eutectic. As the well-depth ratio decreases, the maximum pressure in the solid-liquid-vapor locus increases.

MONTE CARLO SIMULATIONS OF COMPLETE PHASE EQUILIBRIA FOR BINARY MIXTURES

by

MONICA RENEE HITCHCOCK

A thesis submitted to the Graduate Faculty
of North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Chemical Engineering

Raleigh, NC 27695

2000

APPROVED BY:

Carol K. Hall Chair of Advisory Committee	Robert E. Funderlic
Keith E. Gubbins	Peter K. Kilpatrick

Biography

The author was born in Montrose, Pennsylvania on February 10, 1971. She is the elder daughter of Leonard and Lorraine Hitchcock and has one sister, Lesley Hitchcock Tewksbury. She received a B. S. degree in Chemical Engineering from Syracuse University in December, 1993. In January, 1994, she was admitted to North Carolina State University to pursue graduate studies in Chemical Engineering. She received an M. S. degree in Chemical Engineering in December, 1998. On October 7, 2000, she married David Lamm, of Wilson, North Carolina.

Acknowledgements

It is a pleasure to acknowledge the many people who have contributed their wisdom, support, and encouragement to the preparation of this thesis.

First, I thank my thesis advisor, Professor Carol Hall, for her guidance and unfailing confidence in my abilities to conduct scientific research. Professor Hall is extremely committed to advising graduate students and I feel fortunate to have been one of her students.

I take this opportunity to warmly acknowledge Professor Cynthia Hirztel who first inspired me to pursue graduate studies while I was an undergraduate at Syracuse University.

I gratefully acknowledge helpful discussions with Professor David Kofke (State University of New York at Buffalo) who was instrumental in helping me get started with the Gibbs-Duhem integration method. I also thank Professor Cor Peters (Delft University of Technology, The Netherlands), Professor Gerhard Schneider (University of Bochum, Germany), and Professor Leonid Boshkov (Academy of Refrigeration, Ukraine) who promptly answered my questions regarding the topic of phase equilibria.

I thank the GAANN Computational Sciences Fellowship Program of the U.S. Department of Education, the Office of Energy Research, Basic Sciences, Chemical Science Division of the U.S. Department of Energy, and the Donors of the Petroleum Research Fund administered by the American Chemical Society for financial support throughout my stay in graduate school.

I am indebted to the Hall research group's past and present computer system administrators: Harpreet Gulati, Nirupama Kenkare, Julie McCormick, Brian Attwood, and Andrew Schultz. None of the work in presented in this thesis would have been possible without their incredible talent, sacrifice, and hard work.

To the members of my research group, as well as my other colleagues in the department with whom I have interacted, both at work and at play, I thank you for your friendship and for the fond memories of intramural teams, cycling outings, and camping trips you have helped create. Special thanks to my former roommates, Florence Henon and Anne Voegler Smith, for making life outside of school enjoyable.

I am especially thankful for the love and support of family and friends during my graduate school career. My parents, Leonard and Lorraine Hitchcock, have always been a source of love and encouragement and I am forever thankful. I would like to thank my sister, Lesley, her husband, Kevin, and their wonderful children, Susanne, Seth, and Travis, for all the lively, refreshing visits to their home. A very special thank you to the Lamm family and to my friends Kenneth and Ivy Parker, for providing a 'home away from home' in North Carolina. Finally, I thank my husband, David, for always believing in me, always finding a way to make me laugh, and always encouraging me to do my best.

Table of Contents

List of Tables

List of Figures

Page

CHAPTER 1

INTRODUCTION

This thesis describes a study of the phase equilibria of binary mixtures using molecular simulation. Of particular interest to us is equilibria involving a solid phase. Our motivation for studying solid-fluid equilibria in mixtures stems from our interest in chiral drug separations. Many chiral drugs are prepared and sold as an equimolar mixture of both enantiomers; in this case the chiral drug is termed a racemate and the 50/50 mixture is described as a racemic mixture. The enzymes and cell surface receptors in our body are also chiral and usually absorb and degrade the two enantiomers present in a racemic chiral drug at different rates. In fact, the two enantiomers in the drug may not have equal levels of pharmacological activity. It is not unusual for just one of the enantiomers to provide the drug's intended therapeutic function, while the other enantiomer remains inactive, causes side effects, or has a different kind of therapeutic function. The enantiomers of the racemate, ketoprofen, are an example of the latter possibility. (*S*)-Ketoprofen is an analgesic/anti-inflammatory agent and (*R*)-ketoprofen provides no analgesic effect but has been shown to slow bone loss in periodontal disease[1].

Over the past decade, the Food and Drug Administration (FDA) has issued di-

rectives to encourage the pharmaceutical industry to produce and sell single enantiomer chiral drugs rather than the racemic drugs that now exist in the marketplace[1–3]. The rationale behind this directive is multi-fold. First, often only one of the enantiomers in a racemic drug is the potent ingredient. For example, ibuprofen pain-reliever is sold as a racemic mixture of *S*- and *R*-ibuprofen, but the only pharmacologically active ingredient is the *S*-ibuprofen. Second, a single enantiomer drug often performs better than its counterpart racemic drug. Removing the unwanted enantiomer reduces the dosage amount and decreases the time needed for the drug to take effect in the body[4]. Third, side effects are reduced or eliminated completely in many cases by switching from a racemate to a single enantiomer drug. Finally, patients diagnosed with diseases such as arthritis, diabetes, hypertension, and neurological illnesses often take medicines to treat these conditions for the remainder of their lives. Removing the non-effective enantiomer from these drugs reduces the chance of unforeseen health problems caused by extended drug treatment.

To comply with the FDA directive, pharmaceutical companies now include synthesis and separation steps designed to isolate enantiomers whenever they develop a new chiral drug. Many companies are also redesigning their existing processes for making racemic drugs to produce enantiomerically pure counterparts. Despite the large interest in producing single enantiomer drug compounds, the majority of separation schemes have been developed through trial-and-error experimentation[5]. Fundamental research focused on methods for producing enantiomerically pure compounds is needed as the pharmaceutical industry moves towards the pro-

duction of more single enantiomer drugs.

Enantiomers pose a unique separation problem because they have identical physical properties. This means that traditional separation techniques such as distillation, extraction, or crystallization cannot be used to separate an enantiomer mixture. One method for separating racemic mixtures on an industrial scale is diastereomeric crystallization.[4,6,7] In this process, the enantiomers are first converted to diastereomers by chemical reaction with a chiral resolving agent in a solvent. Unlike enantiomers, diastereomers are isomers with different physical properties and are separable by conventional methods. Here the solubility difference between the two diastereomers is utilized and the reaction mixture is cooled to crystallize the least soluble diastereomer from the solution. The resulting diastereomeric solid is then treated by further chemical reactions to remove the resolving agent and retrieve the pure enantiomer.

Currently, there are no theories available to govern the selection of the solvent and chiral resolving agent used to separate a given pair of enantiomers. The ultimate goal of this research is to predict the phase behavior for a mixture of chiral molecules using molecular simulation. Achieving this goal will require the following developments in molecular simulation tools:

- Develop a molecular simulation method to calculate solid-liquid equilibria for binary mixtures of spherical molecules.
- Extend the simulation method to ternary mixtures of spherical molecules.
- Extend the simulation method to non-spherical molecules.

- Apply the method to ternary mixtures consisting of a pair of model diastereomers and a solvent.

The work we present in this thesis represents the first step toward predicting solid-liquid phase diagrams for mixtures of chiral molecules via molecular simulation.

1.1 Overview

In this section, we provide a short summary of the subsequent chapters in this thesis. Chapter 2 has been submitted for publication, Chapter 3 has been accepted for publication, Chapter 4 will be submitted shortly, and Chapter 5 has been accepted for publication. All of the chapters are self-contained units complete with literature review and bibliography.

Chapter 2 explores the effect of both molecular size and intermolecular attractions on the complete phase behavior of a mixture. We calculate complete phase diagrams for binary Lennard-Jones mixtures with diameter ratios ranging from 0.85 – 0.95 and attractive well-depth ratios ranging from 0.625 – 1.6, at a reduced pressure $P^* \equiv P\sigma_{11}^3/\epsilon_{11} = 0.002$, which is equivalent to atmospheric pressure for argon. The cross-species interaction parameters are obtained from Lorentz-Berthelot combining rules. We restrict ourselves to diameter ratios ranging from 0.85 to 0.95 because in this region the only kind of solid phase that can form is a substitutionally disordered fcc solid solution (the two species pack onto the same fcc crystalline lattice and can substitute for one another in any order on the lattice). At diameter ratios less than 0.85 the calculation is more complex because several ordered

crystalline phases are possible, necessitating the calculation of each phase's free energy to determine the most stable crystalline structure. Vapor-liquid, solid-liquid, and solid-vapor lines are calculated for each mixture by integrating the Clapeyron differential equation for binary mixture phase equilibria at constant pressure. The initial conditions for the integrations are the vapor-liquid and solid-liquid coexistence data for a single component Lennard-Jones system at $P^* = 0.002$ obtained via Gibbs-Duhem integration[8,9]. The properties of each phase at subsequent integration points are determined by semigrand canonical Monte Carlo simulations (constant temperature, pressure, total number of molecules, and fugacity fraction) on the vapor, liquid, and solid phases. We find that for well-depth ratios of unity (equal attractions among species) there is no interference between the vapor-liquid and solid-liquid coexistence regions. As the well-depth ratio increases or decreases from unity, the vapor-liquid and solid-liquid phase envelopes widen and interfere with each other leading to a solid-vapor coexistence region. For all well-depth ratios and a diameter ratio of 0.95, the solid-liquid lines have a shape characteristic of a solid solution (with or without a minimum melting temperature); as the diameter ratio decreases the solid-liquid lines fall to lower temperatures until they eventually drop below the solid-solid coexistence region, resulting in either a eutectic or peritectic three-phase line.

Chapter 3 explores the effect of the binary interaction parameter, δ_{12}, on the complete phase behavior of a mixture. We present complete phase diagrams for binary Lennard-Jones mixtures with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$, well-depth ra-

tio $\epsilon_{11}/\epsilon_{22} = 0.45$, and binary interaction parameters $\delta_{12} = 1.0, 0.9,$ and 0.75, at reduced pressure, $P^* = P\sigma_{11}^3/\epsilon_{11} = 0.05$. For the mixture with $\delta_{12} = 1$ we find a completely miscible vapor-liquid coexistence region with a eutectic solid-liquid coexistence region. These two regions are separated by a completely miscible liquid phase. For the mixtures with $\delta_{12} < 1$ we find that the vapor-liquid and solid-liquid coexistence regions interfere. This interference results in a vapor-solid coexistence region bounded above and below by solid-liquid-vapor coexistence lines. We also find that the mixtures with $\delta_{12} < 1$ have a region of liquid-liquid immiscibility that is metastable with respect to the solid-fluid phase equilibria.

Chapter 4 explores the effect of pressure on the complete phase behavior of a mixture. We present complete, $T - x$ phase diagrams for binary Lennard-Jones mixtures with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$ and $\epsilon_{11}/\epsilon_{22} = 1.6$ at reduced pressures $P^* = 0.002$, 0.01, 0.025, 0.05, and 0.1. We find that as pressure increases, the vapor-liquid coexistence region first shifts to higher temperatures and then begins to disappear as the pressure approaches critical conditions. We summarize these results on a $P - T$ projection that identifies the three-phase coexistence features of the mixture (solid-liquid-vapor and solid-solid-liquid) in addition to the pure component vapor-liquid, solid-liquid, and vapor-solid coexistence curves. We then present three more $P - T$ projections for mixtures with $\sigma_{11}/\sigma_{22} = 0.85$, 0.9, and 0.95, and $\epsilon_{11}/\epsilon_{22} = 0.45$ so that we can observe how the features on these phase diagrams change with variations in diameter ratio σ_{11}/σ_{22} and well-depth ratio $\epsilon_{11}/\epsilon_{22}$. We find that as the diameter ratio decreases the maximum in the locus of solid-liquid-

vapor coexistence pressures decreases and the locus of solid(1)-solid(2)-liquid temperatures shifts from temperatures above the solid-liquid temperature of pure component 1 to temperatures below the solid-liquid coexistence temperature of pure component 1. We find that as well-depth ratio decreases the coexistence curves for pure component 2 shift from temperatures and pressures below those of pure component 1 to temperatures and pressures above those of pure component 1 and that the maximum in the locus of solid-liquid-vapor coexistence pressures increases.

Chapter 5 is a study of a binary Lennard-Jones mixture with the following parameters: $\sigma_{11} = \sigma_{22} = 1.0$, $\sigma_{12} = 0.85$, $\epsilon_{11} = 1.0$, $\epsilon_{12} = 0.82$, and $\epsilon_{22} = 1.2$. We calculate complete $T - x$ phase diagrams for this mixture at reduced pressures P^* = 0.007, 0.05, and 0.1. Using these results we construct a $P - T$ projection showing the three-phase loci for heteroazeotropes ($l_1 l_2 v$), monotectics ($l_1 l_2 s$), and eutectics ($s_1 s_2 l$). Fluid phase coexistence points for this mixture have also been calculated by Canongia Lopes[10] using Gibbs ensemble Monte Carlo. They found a locus of heteroazeotrope temperatures that coincides with our heteroazeotrope locus but disappears above $T^* = 1.1$ and below $T^* = 0.9$. Canongia Lopes interpreted this disappearance of liquid-liquid immiscibility above $T^* = 1.1$ and below $T^* = 0.9$ as evidence for the presence of upper and lower critical endpoints, respectively. Based on these inferred upper and lower critical endpoints, Canongia Lopes concluded that this mixture displays Type VI behavior (closed-loop liquid-liquid immiscibility). In our calculations we did not observe any evidence of Type VI phase behavior and instead conclude that this mixture displays Type II phase behavior.

For completeness, an earlier study on solid-liquid equilibria for binary Lennard-Jones mixtures has been included in Appendix A. This work was defended as part of my Master of Science research in October, 1998. To our knowledge, these were the first direct simulations of solid-liquid equilibrium in binary Lennard-Jones mixtures. We calculated solid-liquid phase diagrams for binary Lennard-Jones mixtures with diameter ratios ranging from 0.85 to 1 and attractive well-depth ratios ranging from 0.45 to 1.6, at a reduced pressure $P^* = 0.002$. In a test of the cell theory predictions[11] for model argon-methane, argon-krypton, and krypton-methane systems, we found that cell theory can qualitatively predict the shape of phase diagrams for Lennard-Jones mixtures. Comparison of our simulation results for the argon-krypton Lennard-Jones system with hard sphere results at the same diameter ratio indicated that the presence of attractive interactions can change the type of phase diagram observed from azeotrope (hard sphere) to solid solution (Lennard-Jones). This suggested that attractive interactions are an important factor in determining the type of solid-liquid phase behavior observed and prompted a more thorough investigation of the effect of variations in well-depth ratio on solid-liquid phase diagrams. We simulated 56 binary Lennard-Jones mixtures over the range of parameters listed above to determine how solid-liquid phase diagrams change as a function of diameter ratio and well-depth ratio. We found that for well-depth ratios of unity (equal attractions among species) phase behavior indicative of azeotropes and eutectics is observed for diameter ratios ranging from 0.85 to 1. We then varied the well-depth ratio of the mixtures at several constant diameter ratios and observed

transitions from azeotrope to solid solution, from azeotrope to eutectic, and from solid solution to simple peritectic. Using our simulation results, we were able to map out the boundaries separating regimes of solid solution, azeotrope and eutectic solid-liquid phase behavior in the space spanned by the Lennard-Jones diameter ratio and well-depth ratios.

Chapters 2 through 5 and Appendix A have been adapted from the following publications:

Chapter 2 M. H. Lamm and C. K. Hall, "Molecular simulation of complete phase diagrams for binary mixtures", *AIChE Journal*, (submitted, 2000).

Chapter 3 M. H. Lamm and C. K. Hall, "Monte Carlo simulations of complete phase diagrams for binary Lennard-Jones mixtures", *Fluid Phase Equilibria*, (accepted, 2000).

Chapter 4 M. H. Lamm and C. K. Hall, "The effect of pressure on complete phase behavior for binary mixtures", *Journal of Chemical Physics*, (to be submitted).

Chapter 5 M. R. Hitchcock and C. K. Hall, "Complete phase diagrams for binary mixtures via Gibbs-Duhem integration", *Proceedings of the Foundations of Molecular Modeling and Simulation Conference*, Keystone, CO, 2000, (accepted, 2000).

Appendix A M. R. Hitchcock and C. K. Hall, "Solid-liquid phase diagrams for binary Lennard-Jones mixtures", *Journal of Chemical Physics*, **110**, 11433-11444 (2000).

1.2 References

[1] S. C. Stinson, Chemical and Engineering News, **71**, 38+ (1993).

[2] S. C. Stinson, Chemical and Engineering News, **73**, 44+ (1995).

[3] S. C. Stinson, Chemical and Engineering News, **72**, 38+ (1994).

[4] C. R. Bayley and N. A. Vaidya. in *Chirality in Industry* edited by A. N. CollinsG. N. Sheldrake and J. Crosby, pages 69–77, (1992).

[5] J. Jacques, A. Collet, and S. H. Wilen, *Enantiomers, Racemates, and Resolutions*. (Krieger Publishing Company, Malabar, Florida, 1981).

[6] J. Crosby, Tetrahedron, **47**(27), 4789–4846 (1991).

[7] F. J. J. Leusen, J. H. Noordik, and H. R. Karfunkel, Tetrahedron, **49**(24), 5377–5396 (1993).

[8] D. A. Kofke, Journal of Chemical Physics, **98**(5), 4149–4162 (1993).

[9] R. Agrawal and D. A. Kofke, Molecular Physics, **85**(1), 43–59 (1995).

[10] J. N. C. Lopes, Molecular Physics, **96**(11), 1649–1658 (1999).

[11] X. Cottin and P. A. Monson, Journal of Chemical Physics, **105**(22), 10022–10029 (1996).

CHAPTER 2

MOLECULAR SIMULATION OF COMPLETE PHASE DIAGRAMS FOR BINARY MIXTURES

2.1 Introduction

Knowledge of the phase behavior of mixtures is crucial to the successful design and operation of countless processes encountered in chemical engineering practice. Design engineers need to know, for example, whether they will encounter precipitates in pipelines, liquid-liquid separation in a distillation column, or cosolvency in a supercritical fluid extraction column. Although the phase equilibrium of a mixture can, in principle, be measured at any condition of interest, ultimately, one would like to be able to predict mixture phase behavior based solely upon knowledge of the components' molecular architecture and intermolecular forces.

The prediction of phase equilibria for mixtures has been the subject of intensive investigation for decades. Two kinds of studies have been conducted: (1) those aimed at developing accurate predictions of phase equilibria for specific substances (*e.g.*, hydrocarbon mixtures, polar/non-polar mixtures, etc.), and (2) those aimed

at providing general intuition about the overall topography of phase diagrams for broad classes of substances, with particular focus on how intermolecular forces impact phase diagram shape. The latter type of study is exemplified by the work of van Konynenburg and Scott[1,2], who analyzed how the types of phase diagrams predicted by the van der Waals equation of state for binary mixtures depend on the values of the van der Waals size and energy parameters. Remarkably, the simple van der Waals equation of state was found to exhibit five of the six types of fluid phase behavior observed experimentally. This landmark study has been followed by similar analyses for other equations of state, such as the Redlich-Kwong[3], the Carnahan-Starling-Redlich-Kwong[4], the Guggenheim[5], and the Ree[6] equations of state.

Most of the research aimed at understanding how intermolecular interactions affect phase behavior has focused exclusively on fluid phase equilibria. However in real systems solid phases form and often interrupt the complex phase behavior exhibited by fluids[7,8]. Phenomenological descriptions of complete phase behavior (*i.e.*, showing equilibrium between vapor, liquid, and solid phases) have been given by Luks[9], Peters *et al.*[10], and Valyashko[11,12]. Luks and Peters *et al.* describe four types of complete phase diagrams observed for binary mixtures of solvent (methane, ethane, carbon dioxide) and a homologous series of solutes (*n*-alkanes). Valyashko proposes a classification scheme for twelve types of complete diagrams. Eight of these types result from the analysis of experimental data for water-inorganic salt mixtures. The remaining four types were deduced by the method of continuous

topological transformation, which involves making educated guesses about the transitions in topography between the eight known types. This method is based on the idea that there are continuous transitions between all types of phase behavior[13].

A quantitative description of complete phase behavior has been given by Luks and coworkers[14,15]. Garcia and Luks[14] calculate the solid-liquid-vapor locus for binary mixtures of solvent and a homologous series of solutes using the van der Waals equation of state for the fluid phase and a simple fugacity model for the solid phase. This work was extended by Labadie *et al.*[15] who calculated the fluid phase critical loci for these mixtures, offering a picture of how the multiphase topography progresses with changes in the solute properties. They found examples of solid-fluid phase behavior in keeping with what has been observed in real systems, as well as solid-fluid phase behavior that has yet to be verified by experiment. As with any analytical equation of state, there exists the possibility that the new topographies are mathematical artifacts stemming from the approximations made in the development of the equation of state. Nonetheless, the new possibilities for complete phase behavior calculated by Luks and coworkers are intriguing and invite further investigation.

The most accurate way to determine how molecular size, shape, and energy of interaction influence phase equilibria is with molecular simulation, since molecular simulations provide exact results for the model system being studied[16]. A significant advance was made in the simulation of phase equilibria when Panagiotopoulos[17,18] introduced the Gibbs ensemble method. In this method, the two coexisting phases

are simulated independently, yet are coupled thermodynamically in order to satisfy the criteria of phase equilibrium: equal temperatures, pressures, and species chemical potentials. These conditions are satisfied by simulating each phase at the same temperature and pressure, and performing particle transfers between the phases to maintain chemical potential equality.

Although the Gibbs ensemble method has been widely adopted for studying fluid phase equilibria, it is not an efficient method for studying solid phase equilibria because a large number of successful particle transfers between each phase are required for chemical potential equilibration. Inspired by the Gibbs ensemble method, Kofke[19,20] introduced the Gibbs-Duhem integration technique for direct simulation of phase equilibria. As in the Gibbs ensemble method, two or more coexisting phases are simulated independently at the same temperature and pressure. However, instead of using particle transfers, the chemical potential equality among each phase is maintained by integrating along the Clapeyron differential equation for coexistence during the simulations. Eliminating the need for particle transfers between phases makes the Gibbs-Duhem integration method well-suited for calculating phase equilibrium for cases in which one of the phases is a solid. The method requires an initial coexistence condition to begin the integration of the Clapeyron equation; this initial condition can be obtained from simulation data (*e.g.*, a Gibbs ensemble simulation or a previous Gibbs-Duhem integration), a reliable theory, or experimental data.

The focus of most of the simulations of vapor-liquid and liquid-liquid phase

behavior has been the Lennard-Jones fluid, the quintessential model of a system containing spherically symmetric molecules. The Lennard-Jones intermolecular potential is given by

$$u_{ij}(r) = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r}\right)^{12} - \left(\frac{\sigma_{ij}}{r}\right)^{6}\right], \tag{2.1}$$

where u_{ij} is the potential energy of interaction between particles i and j, r is the distance between particles i and j, ϵ_{ij} is the Lennard-Jones attractive well-depth, and σ_{ij} is the Lennard-Jones diameter. Simulations of the fluid phase behavior for Lennard-Jones mixtures have proven useful for testing theories[21–23] as well as for developing general intuition regarding the influence of molecular size and intermolecular interactions on phase behavior[24–28].

In Appendix A we used the Gibbs-Duhem integration method combined with semigrand canonical Monte Carlo simulations to calculate solid-liquid phase diagrams for binary Lennard-Jones mixtures over a range of diameter ratios $\sigma_{11}/\sigma_{22} = 0.85-1.0$ and well-depth ratios $\epsilon_{11}/\epsilon_{22} = 0.45-1.6$. The cross-species interaction parameters were calculated using the Lorentz-Berthelot[30] combining rules. We found that for well-depth ratios of unity (equal attractions among species), phase behavior indicative of eutectics and solid solutions with minimum melting points is observed for diameter ratios ranging from 0.85 to 1. We then varied the well-depth ratio of the mixtures at several constant diameter ratios and observed transitions from solid solution to solid solution with a minimum melting point, from solid solution with a minimum melting point to eutectic, and from solid solution to peritectic. Using our simulation results, we were able to map out the boundaries separating regimes of

solid solution, solid solution with a minimum melting point, eutectic, and peritectic solid-liquid phase behavior in the space spanned by the Lennard-Jones diameter and well-depth ratios.

More recently, we have demonstrated that the Gibbs-Duhem integration method can be used to calculate complete phase diagrams[31,32] where by "complete" we mean containing all possible phases: solid, liquid, and vapor. Prior to that, complete phase diagrams for symmetric (equal diameters, $\sigma_{11} = \sigma_{22}$; equal attractive well-depths, $\epsilon_{11} = \epsilon_{22}$) Lennard-Jones mixtures were calculated by Vlot *et al.*[33] using a combination of molecular simulation and semiempirical models. In their work, Monte Carlo simulations were conducted for each phase at selected state points to determine the excess free energy as a function of composition. The resulting free energy versus composition data was fit with a two-parameter Redlich-Kister polynomial and the convex envelope construction method was used to determine the phase diagram. In comparison to this somewhat indirect method, the Gibbs-Duhem integration method can be applied to the calculation of complete phase diagrams with relative ease.

Our objective in this chapter is to explore the effect of both molecular size and intermolecular attractions on the complete phase behavior of a mixture. We calculate complete phase diagrams for binary Lennard-Jones mixtures with diameter ratios ranging from 0.85 – 0.95 and attractive well-depth ratios ranging from 0.625 – 1.6, at a reduced pressure $P^* \equiv P\sigma_{11}^3/\epsilon_{11} = 0.002$, which is equivalent to atmospheric pressure for argon. The cross-species interaction parameters are obtained from

Lorentz-Berthelot combining rules. We restrict ourselves to diameter ratios ranging from 0.85 to 0.95 because in this region the only kind of solid phase that can form is a substitutionally disordered fcc solid solution (the two species pack onto the same fcc crystalline lattice and can substitute for one another in any order on the lattice). (At diameter ratios less than 0.85, the calculation is more complex because several ordered crystalline phases are possible, necessitating the calculation of each phase's the free energy to determine the most stable crystalline structure.) Vapor-liquid, solid-liquid, and solid-vapor lines are calculated for each mixture by integrating the Clapeyron differential equation for binary mixture phase equilibria at constant pressure. The initial conditions for the integrations are the vapor-liquid and solid-liquid coexistence data for a single component Lennard-Jones system at $P^* = 0.002$ obtained via Gibbs-Duhem integration[19,34]. The properties of each phase at subsequent integration points are determined by semigrand canonical Monte Carlo simulations (constant temperature, pressure, total number of molecules, and fugacity fraction) on the vapor, liquid, and solid phases.

Highlights of our results are the following. We find that for well-depth ratios of unity (equal attractions among species) there is no interference between the vapor-liquid and solid-liquid coexistence regions. As the well-depth ratio increases or decreases from unity, the vapor-liquid and solid-liquid phase envelopes widen and interfere with each other leading to a solid-vapor coexistence region. For all well-depth ratios and a diameter ratio of 0.95, the solid-liquid lines have a shape characteristic of a solid solution (with or without a minimum melting temperature); as

the diameter ratio decreases the solid-liquid lines fall to lower temperatures until they eventually drop below the solid-solid coexistence region, resulting in either a eutectic or peritectic three-phase line.

The remainder of the chapter is organized as follows. In the next section we outline the Gibbs-Duhem integration method and describe how we applied the procedure to the calculation of mixture phase behavior for solid, liquid, and vapor phases. We then present the complete phase diagrams, followed by a discussion of the results. Finally, we conclude with a brief summary and further discussion.

2.2 Gibbs-Duhem integration

In this section we describe how we calculated phase equilibria for binary Lennard-Jones mixtures using the Gibbs-Duhem integration method. We begin by presenting a brief review of the Gibbs-Duhem integration method. We then discuss our procedures for determining an initial coexistence condition and integrating the Clapeyron equation. Finally, we describe the details of the semigrand ensemble simulations used throughout the integration procedure to determine the properties of each of the coexisting phases.

The coexistence lines were calculated using Gibbs-Duhem integration[20,35]. In this method, phase coexistence is determined by numerically integrating the Clapeyron differential equation appropriate to the system of interest. Clapeyron equations describe how field variables (variables that must be equal among coexisting phases) change along the phase equilibrium line. The Clapeyron equation for equilibrium

between two phases (α and γ) of a binary mixture containing components 1 and 2 at constant pressure is

$$\frac{d\beta}{d\xi_2} = \frac{(x_2^\alpha - x_2^\gamma)}{\xi_2(1-\xi_2)(h^\alpha - h^\gamma)}, \tag{2.2}$$

where β is the reciprocal temperature, $1/kT$, with k the Boltzmann constant and T the absolute temperature, ξ_2 is the fugacity fraction of species 2, $\xi_2 \equiv \hat{f}_2 / \sum \hat{f}_i$, with $\hat{f}_i$, the fugacity of species i in solution, x_2 is the mole fraction of species 2, and h is the molar enthalpy. The right-hand side of Eq. (2.2) can be integrated numerically to find an equation for β as a function of ξ_2 if we have an initial condition describing the temperature, fugacity fraction, enthalpies and compositions at one coexistence point.

2.2.1 Initial condition

An initial coexistence condition is necessary to begin a Gibbs-Duhem integration calculation for phase equilibrium in a binary mixture. A convenient choice for the initial coexistence condition is the vapor-liquid or solid-liquid equilibrium condition for either of the pure Lennard-Jones components. Here, we used literature data obtained via Gibbs-Duhem integration for the vapor-liquid[19] and solid-liquid[34] coexistence conditions. The integrand in Eq. (2.2) is undefined for pure components ($\xi_2 = 0, x_2 = 0$ and $\xi_2 = 1, x_2 = 1$) but it can be estimated using the limiting case of infinite dilution. Here we follow Mehta and Kofke[36], who used the infinite dilution case to start their Gibbs-Duhem integration calculations of vapor-liquid equilibria in binary mixtures.

The limiting value of the integrand when x_2 approaches zero, $(d\beta/d\xi_2)_{x_2=0}$, can be estimated by supposing that the real mixture displays ideal solution behavior at the limit of infinite dilution of species 2. With this assumption, the abundant component (species 1) in the ideal solution follows the Lewis Randall rule

$$\hat{f}_1 = x_1 f_1, \tag{2.3}$$

while the dilute component (species 2) obeys Henry's law

$$\hat{f}_2 = x_2 H_2, \tag{2.4}$$

where f_1 is the fugacity of pure component 1 at the temperature and pressure of the mixture, and H_2 is the Henry's law constant for species 2. Letting $x_1 \to 1$ and $x_2 \to 0$, the fugacity fraction of species 2 becomes, $\xi_2 = x_2 H_2 / f_1$. After making these substitutions into Eq. (2.2) and rearranging terms[29] we get

$$\left.\frac{d\beta}{d\xi_2}\right|_{x_2=0} = \frac{f_1(1/H_2^{\alpha} - 1/H_2^{\gamma})}{(h^l - h^s)}. \tag{2.5}$$

This gives us an estimate for the integrand at the initial condition of $\xi_2 = 0$ and coexistence (*e.g.,* solid-liquid, vapor-liquid) temperature, T_1, of pure species 1. A similar formula can be derived in the limit of infinite dilution of species 1.

We can calculate all of the quantities on the right-hand side of Eq. (2.5) with an NPT simulation of pure species 1. The molar enthalpy of each phase is h =

$\langle u + Pv \rangle_{\mathrm{NPT}}$, where u is the configurational energy of the system and the brackets, $\langle\rangle_{\mathrm{NPT}}$, denote an NPT ensemble average. The quantity f_1/H_2 can be calculated from[36]

$$\frac{f_1}{H_2} = \langle \exp(-\beta \Delta u_{1\to 2}) \rangle_{\mathrm{NPT}} \tag{2.6}$$

where $\Delta u_{1\to 2}$ is the exchange energy associated with switching a particle from species 1 to 2. The exchange energy, $\Delta u_{1\to 2}$, can be obtained by conducting trial identity switches during the simulation; these involve randomly selecting a particle and calculating the energy that would result if we were to switch the particle from species 1 to species 2. This is done without actually changing the particle's identity.

2.2.2 Integration

Once we have an initial coexistence condition, the Gibbs-Duhem integration procedure may be performed over the entire range of fugacity fractions, $\xi_2 = 0$ to $\xi_2 = 1$, using a predictor-corrector algorithm to integrate Eq. (2.2). Starting at the initial condition, $\beta_0, (\xi_2)_0$, we step to the next fugacity fraction, $(\xi_2)_1$, and estimate the associated reciprocal temperature, β_1, using the trapezoid-rule predictor formula

$$\beta_1^{(0)} = \beta_0 + [(\xi_2)_1 - (\xi_2)_0] F(\beta_0, (\xi_2)_0), \tag{2.7}$$

where the superscript "0" indicates that $\beta_1^{(0)}$ (a predicted value) is our zeroth iteration attempt at finding the reciprocal temperature β_1 and F is the right-hand side of Eq. (2.2) evaluated at the initial condition. Once $\beta_1^{(0)}$ is estimated at the given

fugacity fraction $(\xi_2)_1$, two semigrand canonical (NPTξ_2) Monte Carlo simulations (one for the α phase and one for the γ phase) are conducted in order to calculate the enthalpies and mole fractions of each phase at the new state point. (Details of the NPTξ_2 simulations will be given in Section 2.2.3.)

After the enthalpies and mole fractions at the new state point are calculated we refine the estimate for β_1 at $(\xi_2)_1$ by performing a loop of corrector iterations until β_1 converges within an acceptable tolerance. The general form of the trapezoid-rule corrector for this loop is given by

$$\beta_1^{(i+1)} = \beta_0 + \frac{[(\xi_2)_1 - (\xi_2)_0]}{2}\left[F_1^{(i)}(\beta_1^{(i)}, (\xi_2)_1) + F_0(\beta_0, (\xi_2)_0)\right], \tag{2.8}$$

where the superscripts (i) and $(i+1)$ denote the iterations of the corrector, the subscripts "0" and "1" denote the initial and current state point, respectively, and $F_1^{(i)}$ is calculated from simulation averages of the enthalpies and mole fractions during the i^{th} iteration of the corrector at $\beta_1^{(i)}$ and $(\xi_2)_1$. After β_1 converges a production segment of simulations are run to obtain the final average enthalpies and mole fractions for the coexistence point. Once the production runs are completed, the fugacity fraction is incremented and the predictor-corrector algorithm described above is repeated to obtain the next state point, β_2, $(\xi_2)_2$.

Higher order predictor-corrector equations are used as we obtain more state points. The midpoint predictor-corrector is used once two state points are known,

$$\beta_{n+1}^{(0)} = \beta_{n-1} + 2[(\xi_2)_{n+1} - (\xi_2)_n]\, F_n(\beta_n, (\xi_2)_n), \tag{2.9}$$

$$\beta_{n+1}^{(i+1)} = \beta_{n-1} + \frac{[(\xi_2)_{n+1} - (\xi_2)_n]}{3}\left[F_{n+1}^{(i)} + 4F_n + F_{n-1}\right], \tag{2.10}$$

and the modified Adams predictor-corrector[37] is used once three or more state points are known,

$$\beta_{n+1}^{(0)} = \beta_n + \frac{[(\xi_2)_{n+1} - (\xi_2)_n]}{24}[55F_n - 59F_{n-1} + 37F_{n-2} - 9F_{n-3}], \tag{2.11}$$

$$\beta_{n+1}^{(i+1)} = \beta_n + \frac{[(\xi_2)_{n+1} - (\xi_2)_n]}{24}\left[9F_{n+1}^{(i)} + 19F_n - 5F_{n-1} + F_{n-2}\right]. \tag{2.12}$$

In these sets of equations the predictor is listed first and the corrector second. The subscripts denote the coexistence state points with $(n+1)$ being the current state point and the superscripts denote the iterations of the corrector for the current coexistence state point. By repeating the predictor-corrector algorithm from $\xi_2 = 0$ to $\xi_2 = 1$ we can map out the entire temperature versus composition phase diagram.

In some of the mixtures, we encountered interference between two different two-phase coexistence regions *e.g.*, the vapor-liquid and solid-liquid coexistence curves overlapped, resulting in a three-phase solid-liquid-vapor coexistence line. In this case, two Gibbs-Duhem integrations were conducted, the first starting from the vapor-liquid coexistence temperature for one of the pure components and the second starting from the solid-liquid coexistence temperature for the same pure component. For example, in Figure 2.1 vapor-liquid and solid-liquid coexistence curves can be calculated starting from the vapor-liquid and solid-liquid coexistence temperatures (T_2^{lv} and T_2^{ls}, respectively) of pure component 2. At some ξ_2 (unknown at the commencement of the two integrations) the liquid phase coexistence lines from each

set of coexistence curves will cross, thus, determining the temperature, fugacity fraction, and coexistence compositions of the three coexisting phases: vapor, liquid, and solid. In Figure 2.1 the liquid lines cross at T^{vls} and x_2^l. Below T^{vls} the inner vapor-liquid and solid-liquid coexistence curves (shown by short-dashed and long-dashed lines, respectively) are metastable with respect to the outer vapor-solid coexistence curves, according to the boundary curvature rule[38,39], which states that the boundaries of one-phase regions must meet at a three-phase line with curvatures such that the boundaries extrapolate into the two-phase coexistence regions.. The vapor-solid coexistence curve is determined by a Gibbs-Duhem integration starting from T^{vls} at x_2^v and x_2^s. Other types of three-phase coexistence lines (heteroazeotropes, eutectics, etc.) can be determined in a similar manner.

2.2.3 Simulations

The enthalpies and mole fractions needed as input to the integration of Eq. (2.2) are obtained from semigrand canonical (constant NPTξ_2) Monte Carlo computer simulations[40]. In this work, all simulations were run with a system size of 500 particles at a reduced pressure $P^* = 0.002$. The temperature and fugacity fraction were fixed at the values specified by the Gibbs-Duhem integration predictor-corrector algorithm. There are three types of Monte Carlo trial moves in semigrand canonical simulations: particle displacements, volume change moves, and particle identity exchanges. The particle displacements and volume change moves are conducted just as they are in a standard NPT simulation[16]. In the particle identity exchange moves

a particle is selected at random and given a trial species identity switch, which is accepted according to the ratio of the species fugacity fractions, ξ_1 and ξ_2. The overall acceptance probability[36] for the moves in the NPTξ_2 ensemble is $\min[1, \exp(\Lambda)]$ where

$$\Lambda = -\beta(U^{\text{trial}} - U^{\text{old}}) - \beta P(V^{\text{trial}} - V^{\text{old}}) + N \ln \frac{V^{\text{trial}}}{V^{\text{old}}} + m \ln \frac{\xi_2}{1 - \xi_2}. \qquad (2.13)$$

In Eq. (2.13), U^{trial} and U^{old}, and V^{trial} and V^{old}, are the configurational energies and volumes of the trial and existing states, respectively, $m = +1$ if the trial identity switch is from species 1 to 2, and $m = -1$ if the trial identity switch is from species 2 to 1. In NPTξ_2 simulations the choice of the type of Monte Carlo move is made randomly but weighted such that the ratio of attempted moves is 1 volume change to N particle displacements to N identity switches. The length of the simulation is given in cycles, where one cycle represents 1 volume change attempt, N displacement attempts, and N identity switch attempts. In our work, a typical NPTξ_2 simulation is equilibrated for 3000 cycles and then followed by a production run of 5000 cycles to compute the average enthalpy and mole fraction. The only difference between fluid and solid phase simulations is that to maintain an fcc crystalline structure in the solid phase simulations we impose a single occupancy constraint[41,42] on the trial displacements of particles in the solid, *i.e.,* any displacements that put the particle outside its lattice cell are rejected.

Other details of the NPTξ_2 simulations are as follows. The simulation volume is a cubic box with periodic boundary conditions. The particles interact via the

Lennard-Jones potential model. We determine the cross-species interaction parameters $(\sigma_{12}, \epsilon_{12})$ by using the Lorentz-Berthelot[30] mixing rules $\sigma_{12} = (\sigma_{11} + \sigma_{22})/2$ and $\epsilon_{12} = \sqrt{\epsilon_{11}\epsilon_{22}}$. The potential interactions are truncated at a cutoff radius of half the box length. To compensate for this truncation, a long range correction is applied to the energy calculations during the simulation by assuming a uniform density distribution beyond the cutoff radius[16].

2.3 Results and Discussion

In this section, we present the results of our Gibbs-Duhem integration calculations of complete phase behavior for binary Lennard-Jones mixtures. All of the phase diagrams were calculated at reduced pressure, $P^* \equiv P\sigma_{11}^3/\epsilon_{11} = 0.002$, which is equivalent to atmospheric pressure for argon. We calculated nine phase diagrams for diameter ratios $\sigma_{11}/\sigma_{22} = 0.85$, 0.9, and 0.95, and well-depth ratios $\epsilon_{11}/\epsilon_{22} = 0.625$, 1.0, and 1.6. For these diameter ratios, the solid phase has a substitutionally disordered fcc crystalline structure.

In the first series, we calculated phase diagrams for binary mixtures with $\sigma_{11}/\sigma_{22} = 0.95$ and $\epsilon_{11}/\epsilon_{22} = 0.625$, 1.0, and 1.6. Figure 2.2 shows the temperature-composition phase diagrams obtained via Gibbs-Duhem integration (row 1) along with not-to-scale schematic diagrams (row 2) drawn to illustrate the smaller features of the actual phase diagram more clearly. On the phase diagram for $\epsilon_{11}/\epsilon_{22} = 0.625$, vapor-liquid coexistence lines originate from pure component 2 ($x_2 = 1, T^* \equiv kT/\epsilon_{11} = 1.141$) and decrease in temperature with decreasing fugacity fraction ξ_2.

Solid-liquid coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 1.099$) and decrease in temperature with decreasing fugacity fraction ξ_2. The liquid-vapor and solid-liquid curves meet at $T^* = 1.093$ and form a three-phase, solid-liquid-vapor equilibrium line. Solid-vapor coexistence lines originate from this three-phase line and decrease in temperature with decreasing fugacity fraction ξ_2. Vapor-liquid curves originate from pure component 1 ($x_2 = 0, T^* = 0.732$) and increase in temperature with increasing fugacity fraction ξ_2. The vapor-liquid and vapor-solid curves meet at $T^* = 0.745$ and form a three-phase, vapor-liquid-solid equilibrium line. Liquid-solid coexistence lines originate from this three-phase line and decrease in temperature with decreasing fugacity fraction ξ_2 until they reach the solid-liquid coexistence temperature for pure component 1 ($x_2 = 0, T^* = 0.687$). A miscible solid phase exists below the solid-liquid curves.

On the phase diagram for $\epsilon_{11}/\epsilon_{22} = 1.0$, vapor-liquid coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 0.742$) and decrease in temperature with decreasing fugacity fraction until they reach the vapor-liquid coexistence temperature for pure component 1 ($x_2 = 0, T^* = 0.732$). A miscible liquid phase exists below the vapor-liquid curves. Solid-liquid coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 0.687$) and decrease in temperature with decreasing fugacity fraction ξ_2 to a minimum melting point ($x_2 = 0.508, T^* = 0.666$). The solid-liquid lines then increase in temperature with decreasing fugacity fraction ξ_2 until they reach the solid-liquid coexistence temperature for pure component 1 ($x_2 = 0, T^* = 0.687$). A miscible solid phase exists below the solid-liquid curves.

On the phase diagram for $\epsilon_{11}/\epsilon_{22} = 1.6$, liquid-vapor coexistence lines originate from pure component 1 ($x_2 = 0, T^* = 0.732$) and decrease in temperature with increasing fugacity fraction ξ_2. Solid-liquid coexistence lines originate from pure component 1 ($x_2 = 0, T^* = 0.687$) and decrease in temperature with increasing fugacity fraction ξ_2. The liquid-vapor and solid-liquid curves meet at $T^* = 0.681$ and form a three-phase, solid-liquid-vapor equilibrium line. Solid-vapor coexistence lines originate from this three-phase line and decrease in temperature with increasing fugacity fraction ξ_2. Liquid-vapor coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 0.485$) and increase in temperature with decreasing fugacity fraction ξ_2. The liquid-vapor and solid-vapor curves meet at $T^* = 0.495$ and form another three-phase, solid-liquid-vapor equilibrium line. Solid-liquid lines originate from this three-phase line and decrease in temperature with increasing fugacity fraction ξ_2 until they reach the solid-liquid coexistence temperature for pure component 2 ($x_2 = 0, T^* = 0.429$). A miscible solid phase exists below the solid-liquid curves.

In the second series, we calculated phase diagrams for binary mixtures with $\sigma_{11}/\sigma_{22} = 0.9$ and $\epsilon_{11}/\epsilon_{22} = 0.625$, 1.0, and 1.6. Figure 2.3 shows the temperature-composition phase diagrams obtained via Gibbs-Duhem integration (row 1) along with not-to-scale schematic diagrams (row 2) drawn to illustrate the smaller features of the actual phase diagram more clearly. On the phase diagram for $\epsilon_{11}/\epsilon_{22} = 0.625$, vapor-liquid coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 1.157$) and decrease in temperature with decreasing fugacity fraction ξ_2. Liquid-solid coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 1.099$) and decrease in

temperature with decreasing fugacity fraction ξ_2. The vapor-liquid and liquid-solid curves meet at $T^* = 1.086$ and form a three-phase, vapor-liquid-solid equilibrium line. Vapor-solid coexistence lines originate from this three-phase line and decrease in temperature with decreasing fugacity fraction. Vapor-liquid curves originate from pure component 1 ($x_2 = 0, T^* = 0.732$) and increase in temperature with increasing fugacity fraction. The vapor-liquid and vapor-solid curves meet at $T^* = 0.770$ and form another three-phase, vapor-liquid-solid equilibrium line. Liquid-solid coexistence lines originate from this three-phase line and decrease in temperature with decreasing fugacity fraction ξ_2 to a minimum melting point ($x_2 = 0.195, T^* = 0.665$). The solid-liquid lines then increase in temperature with decreasing fugacity fraction ξ_2 until they reach the solid-liquid coexistence temperature for pure component 1 ($x_2 = 0, T^* = 0.687$). A miscible solid phase exists below the solid-liquid curves.

On the phase diagram for $\epsilon_{11}/\epsilon_{22} = 1.0$, vapor-liquid coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 0.753$) and decrease in temperature with decreasing fugacity fraction until they reach the vapor-liquid coexistence temperature for pure component 1 ($x_2 = 0, T^* = 0.732$). A miscible liquid phase exists below the vapor-liquid curves. Solid-liquid coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 0.687$) and decrease in temperature with decreasing fugacity fraction ξ_2 to a minimum melting point ($x_2 = 0.440, T^* = 0.593$). The solid-liquid lines then increase in temperature with decreasing fugacity fraction ξ_2 until they reach the solid-liquid coexistence temperature for pure component 1 ($x_2 = 0, T^* = 0.687$). A miscible solid phase exists below the solid-liquid curves.

On the phase diagram with $\epsilon_{11}/\epsilon_{22} = 1.6$, liquid-vapor coexistence lines originate from pure component 1 ($x_2 = 0, T^* = 0.732$) and decrease in temperature with increasing fugacity fraction. Solid(1)-liquid coexistence lines originate from pure component 1 ($x_2 = 0, T^* = 0.687$) and decrease in temperature with increasing fugacity fraction ξ_2. The liquid-vapor and solid(1)-liquid curves meet at $T^* = 0.679$ and form a three-phase, solid(1)-liquid-vapor equilibrium line. Solid(1)-vapor coexistence lines originate from this three-phase line and decrease in temperature with increasing fugacity fraction ξ_2. Liquid-vapor coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 0.494$) and increase in temperature with decreasing fugacity fraction ξ_2. The liquid-vapor and solid(1)-vapor curves meet at $T^* = 0.505$ and form another three-phase, solid(1)-liquid-vapor equilibrium line. Solid(1)-liquid lines originate from this three-phase line and decrease in temperature with increasing fugacity fraction ξ_2. Liquid-solid(2) coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 0.429$) and increase in temperature with decreasing fugacity fraction ξ_2. The solid(1)-liquid and liquid-solid(2) curves meet at $T^* = 0.437$ and form a three phase, solid(1)-solid(2)-liquid equilibrium line. This line is also known as a peritectic. Below this temperature (not shown in Figure 2.3) solid(1) and solid(2) are in equilibrium. Although Gibbs-Duhem integration method we employ can be readily applied to solid-solid equilibria we have not calculated solid(1)-solid(2) coexistence lines in this work since they are not necessary for classifying the solid-liquid behavior.

In the third series, we calculated phase diagrams for binary mixtures with

$\sigma_{11}/\sigma_{22} = 0.85$ and $\epsilon_{11}/\epsilon_{22} = 0.625$, 1.0, and 1.6. Figure 2.4 shows the temperature-composition phase diagrams obtained via Gibbs-Duhem integration (row 1) along with not-to-scale schematic diagrams (row 2) drawn to illustrate the smaller features of the actual phase diagram more clearly. On the phase diagram for $\epsilon_{11}/\epsilon_{22} = 0.625$, vapor-liquid coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 1.173$) and decrease in temperature with decreasing fugacity fraction ξ_2. Liquid-solid(2) coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 1.099$) and decrease in temperature with decreasing fugacity fraction. The vapor-liquid and liquid-solid(2) curves meet at $T^* = 1.075$ and form a three-phase, vapor-liquid-solid(2) equilibrium line. Vapor-solid(2) coexistence lines originate from this three-phase line and decrease in temperature with decreasing fugacity fraction ξ_2. Vapor-liquid curves originate from pure component 1 ($x_2 = 0, T^* = 0.732$) and increase in temperature with increasing fugacity fraction ξ_2. The vapor-liquid and vapor-solid(2) curves meet at $T^* = 0.819$ and form another three-phase, vapor-liquid-solid(2) equilibrium line. Liquid-solid(2) coexistence lines originate from this three-phase line and decrease in temperature with decreasing fugacity fraction ξ_2. Solid(1)-liquid coexistence lines originate from pure component 1 ($x_2 = 0, T^* = 0.687$) and decrease in temperature with increasing fugacity fraction ξ_2. The solid(1)-liquid and liquid-solid(2) curves meet at $T^* = 0.543$ and form a three phase, solid(1)-liquid-solid(2) equilibrium line. This line is also known as a eutectic. Below this temperature solid(1) and solid(2) are in equilibrium (not shown).

On the phase diagram for $\epsilon_{11}/\epsilon_{22} = 1.0$, vapor-liquid coexistence lines origi-

nate from pure component 2 ($x_2 = 1, T^* = 0.772$) and decrease in temperature with decreasing fugacity fraction ξ_2 until they reach the vapor-liquid coexistence temperature for pure component 1 ($x_2 = 0, T^* = 0.732$). A miscible liquid phase exists below the vapor-liquid curves. Solid(1)-liquid coexistence lines originate from pure component 1 ($x_2 = 0, T^* = 0.687$) and decrease in temperature with increasing fugacity fraction ξ_2. Liquid-solid(2) coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 0.687$) and decrease in temperature with decreasing fugacity fraction ξ_2. The solid(1)-liquid and liquid-solid(2) curves meet at $T^* = 0.466$ and form a eutectic. Below this temperature solid(1) and solid(2) are in equilibrium (not shown).

On the phase diagram for $\epsilon_{11}/\epsilon_{22} = 1.6$, liquid-vapor coexistence lines originate from pure component 1 ($x_2 = 0, T^* = kT/\epsilon_{11} = 0.732$) and decrease in temperature with increasing fugacity fraction ξ_2. Solid(1)-liquid coexistence lines originate from pure component 1 ($x_2 = 0, T^* = 0.687$) and decrease in temperature with increasing fugacity fraction ξ_2. The liquid-vapor and solid(1)-liquid curves meet at $T^* = 0.6785$ and form a three-phase, solid(1)-liquid-vapor equilibrium line. Solid(1)-vapor coexistence lines originate from this three-phase line and decrease in temperature with increasing fugacity fraction ξ_2. Liquid-vapor coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 0.505$) and increase in temperature with decreasing fugacity fraction ξ_2. The liquid-vapor and solid(1)-vapor curves meet at $T^* = 0.516$ and form another three-phase, solid(1)-liquid-vapor equilibrium line. Solid(1)-liquid lines originate from this three-phase line and decrease in temperature with increasing fugacity fraction ξ_2. Liquid-solid(2) coexistence lines originate from pure com-

ponent 2 ($x_2 = 1, T^* = 0.429$) and decrease in temperature with decreasing fugacity fraction ξ_2. The solid(1)-liquid and liquid-solid(2) curves meet at $T^* = 0.420$ and form a eutectic. Below this temperature solid(1) and solid(2) are in equilibrium (not shown).

To understand how variations in the size and the attractive interactions lead to the complete phase diagrams displayed in Figures 2.2 – 2.4, it is helpful to consider the schematic diagrams shown in Figure 2.5. The columns correspond to mixtures with diameter ratios $\sigma_{11}/\sigma_{22} = 0.95$, 0.9, and 0.85, and the rows correspond to mixtures with well-depth ratios $\epsilon_{11}/\epsilon_{22} = 1.6$, 1.0, and 0.625. The equilibrium curves for vapor-liquid, solid-liquid, and solid-vapor coexistence (black solid lines) are based on our simulation results. The equilibrium curves for solid-solid coexistence (gray solid lines) are based on our best guess as to how the upper critical solution temperature of the solid-solid immiscibility dome shifts with variations in well-depth ratio. Underpinning this guess is quasichemical theory[38,39] which tells us that the upper critical solution temperature increases as the attractions between like molecules become stronger than the attractions between unlike molecules, *i.e.*, for a constant σ_{11}/σ_{22} the upper critical solution temperature will increase as $\epsilon_{11}/\epsilon_{22}$ increases above unity or decreases below unity and for a constant $\epsilon_{11}/\epsilon_{22}$ the upper critical solution temperature will increase as σ_{11}/σ_{22} decreases. The metastable solid-liquid and vapor-liquid (black dashed lines) and the metastable solid-solid (gray dashed lines) coexistence curves are shown to illustrate how these various two-phase coexistence regions interfere as diameter ratio and well-depth ratio are varied.

In the σ_{11}/σ_{22} = 0.95 column, at $\epsilon_{11}/\epsilon_{22}$ = 1.0 there are three, non-interfering two-phase coexistence regions: vapor-liquid, solid-liquid, and solid-solid. The solid-liquid coexistence region has a shape characteristic of a solid solution with a minimum melting point; this minimum melting point is inevitable since the melting temperatures of the pure components are equal[30]. As the well-depth ratio increases to $\epsilon_{11}/\epsilon_{22}$ = 1.6, the vapor-liquid and solid-liquid coexistence temperatures of pure component 2 decrease and the vapor-liquid and solid-liquid phase envelopes widen, causing them to interfere and form a vapor-solid coexistence region. Additionally, at $\epsilon_{11}/\epsilon_{22}$ = 1.6, the difference between the melting temperatures of the pure components is sufficiently large that the minimum in the solid-liquid coexistence curve disappears and the shape of the solid-liquid coexistence region changes from a solid solution with a minimum melting point ($\epsilon_{11}/\epsilon_{22}$ = 1.0) to a simple solid solution ($\epsilon_{11}/\epsilon_{22}$ = 1.6). Likewise, as the well-depth ratio decreases to $\epsilon_{11}/\epsilon_{22}$ = 0.625, the vapor-liquid and solid-liquid coexistence temperatures of pure component 2 increase, resulting in a similar pattern of vapor-liquid, solid-liquid interference and a change in the shape of the solid-liquid coexistence region from a solid solution with a minimum melting point ($\epsilon_{11}/\epsilon_{22}$ = 1.0) to a simple solid solution ($\epsilon_{11}/\epsilon_{22}$ = 0.625).

In the σ_{11}/σ_{22} = 0.90 column, at $\epsilon_{11}/\epsilon_{22}$ = 1.0 there are again three, non-interfering two-phase coexistence regions: vapor-liquid, solid-liquid, and solid-solid. The solid-liquid coexistence region has a shape characteristic of a solid solution with a minimum melting point. As the well-depth ratio increases to $\epsilon_{11}/\epsilon_{22}$ = 1.6, the vapor-liquid and solid-liquid coexistence temperatures of pure component 2 de-

crease and the vapor-liquid and solid-liquid phase envelopes widen, causing them to interfere and form a vapor-solid coexistence region. Additionally, at $\epsilon_{11}/\epsilon_{22} = 1.6$, the difference between the melting temperatures of the pure components is sufficiently large that the minimum in the solid-liquid coexistence curve disappears, however, instead of forming a simple solid solution shape (as was the case for $\sigma_{11}/\sigma_{22} = 0.95$, $\epsilon_{11}/\epsilon_{22} = 1.6$) the solid-liquid coexistence region displays a shape characteristic of a peritectic. This peritectic shape can be thought of as resulting from the interference between a solid-liquid coexistence region with a solid solution shape and a solid-solid immiscibility dome, as shown by the dashed curves. Likewise, as the well-depth ratio decreases to $\epsilon_{11}/\epsilon_{22} = 0.625$, the vapor-liquid and solid-liquid coexistence temperatures of pure component 2 increase, resulting in a similar pattern of vapor-liquid, solid-liquid interference. Here the solid-liquid coexistence region retains its solid solution with a minimum melting temperature shape.

In the $\sigma_{11}/\sigma_{22} = 0.85$ column, at $\epsilon_{11}/\epsilon_{22} = 1.0$ there is a non-interfering vapor-liquid coexistence region followed at lower temperatures by a solid-liquid coexistence region with a shape characteristic of a eutectic. This eutectic shape can be thought of as resulting from the interference between a solid-liquid coexistence region with a minimum melting temperature and a solid-solid immiscibility dome, as shown by the dashed curves. As the well-depth ratio increases to $\epsilon_{11}/\epsilon_{22} = 1.6$, the vapor-liquid and solid-liquid coexistence temperatures of pure component 2 decrease and the vapor-liquid and solid-liquid phase envelopes widen, causing them to interfere and form a vapor-solid coexistence region. The shape of the solid-liquid

coexistence region remains the same, although the eutectic composition (the point at which the two liquid phase lines cross) shifts towards pure component 2. Likewise, as the well-depth ratio decreases to $\epsilon_{11}/\epsilon_{22} = 0.625$, the vapor-liquid and solid-liquid coexistence temperatures of pure component 2 increase, resulting in a similar pattern of vapor-liquid, solid-liquid interference. Again, the shape of the solid-liquid coexistence region remains the same, but here the eutectic composition shifts towards pure component 1.

In the $\epsilon_{11}/\epsilon_{22} = 1.0$ row, there is no interference between the vapor-liquid and solid-liquid coexistence regions. As σ_{11}/σ_{22} decreases, the vapor-liquid coexistence temperature of pure component 2 increases and the vapor-liquid phase envelope widens. At $\sigma_{11}/\sigma_{22} = 0.95$, there is a minimum melting temperature in the solid-liquid coexistence region. As σ_{11}/σ_{22} decreases the solid-liquid lines "fall" because the liquid phase can accommodate the larger differences in size easier than the solid phase[43]; this results in the minimum melting temperature decreasing with decreasing σ_{11}/σ_{22}. At $\sigma_{11}/\sigma_{22} = 0.85$ the minimum melting temperature is below the upper critical solution temperature of the solid-solid coexistence region and a eutectic line is formed. Similar trends can be observed in the $\epsilon_{11}/\epsilon_{22} = 0.625$ and 1.6 rows.

2.4 Summary

The Gibbs-Duhem integration technique was combined with semigrand canonical Monte Carlo simulations to calculate complete $T - x$ phase diagrams for binary

Lennard-Jones mixtures. To explore the effect of molecular size and intermolecular attractions on the complete phase behavior of a mixture, we calculated phase diagrams for binary Lennard-Jones mixtures with diameter ratios ranging from 0.85 – 0.95 and attractive well-depth ratios ranging from 0.625 – 1.6, at a reduced pressure $P^* = 0.002$.

These calculations mark the first time that molecular simulation has been used to obtain phase diagrams describing all types of equilibria between vapor, liquid, and solid phases. This capability affords the possibility of classifying phase diagrams based not only on fluid phase behavior, as was done by Van Konynenburg and Scott, but also on solid phase behavior. The Gibbs-Duhem integration approach holds a few advantages over the equation of state approach for calculating phase equilibria: (1) predictions made by an equation of state are subject to spurious error as a result of approximations, while the Gibbs-Duhem integration predictions are based on simulation results, which are exact for the particular model being studied, and (2) equations of state do not allow for solid phase formation but a solid phase with any crystalline structure can be readily incorporated into the Gibbs-Duhem integration scheme. Finally, the limitations of this approach should be pointed out as well. First, the Gibbs-Duhem integration procedure, as formulated here, cannot be used to evaluate critical phenomena. Second, it is not always trivial to obtain an initial condition to begin the integration. For example, there is presently no straightforward way to calculate liquid-liquid and solid-solid coexistence lines that are not connected (by way of interference with a vapor-liquid or solid-liquid coexistence re-

gion) to a pure component coexistence point. In this case, one would have to conduct several exploratory Gibbs-Duhem integrations along constant pressure and constant temperature paths to obtain an initial coexistence point. Calculations of this nature are currently underway.

2.5 References

[1] R. L. Scott and P. H. van Konynenburg, Discussions of the Faraday Society, **49**, 87–97 (1970).

[2] P. H. van Konynenburg and R. L. Scott, Philosophical Transactions of the Royal Society of London A., **298**, 495–540 (1980).

[3] U. K. Deiters and I. L. Pegg, Journal of Chemical Physics, **90**, 6632–6641 (1989).

[4] T. Kraska and U. K. Deiters, Journal of Chemical Physics, **96**, 539–547 (1992).

[5] J. Wang, G. W. Wu, and R. J. Sadus, Molecular Physics, **98**, 715–723 (2000).

[6] V. A. Mazur, L. Z. Boshkov, and V. G. Murakhovsky, Physics Letters, **104**, 415–417 (1984).

[7] G. M. Schneider, in *Chemical Thermodynamics* edited by M. L. McGlashan (The Chemical Society, London, 1978).

[8] R. L. Scott, Accounts of Chemical Research, **20**, 97–107 (1987).

[9] K. D. Luks. In *Proceedings of the 2nd International Conference on Phase Equilibria and Fluid Properties in the Chemical Industry*, number 11 in EFCE, pages 699–711, Berlin, (1980).

[10] C. J. Peters, R. N. Lichtenthaler, and J. de Swaan Arons, Fluid Phase Equilibria, **29**, 495–504 (1986).

[11] V. M. Valyashko, Z. Phys. Chemie, Leipzig, **267**(3), 481–493 (1986).

[12] V. M. Valyashko, Pure and Applied Chemistry, **62**(11), 2129–2138 (1990).

[13] G. M. Schneider, Advances in Chemical Physics, **17**, 1–42 (1970).

[14] D. C. Garcia and K. D. Luks, Fluid Phase Equilibria, **161**, 91–106 (1999).

[15] J. A. Labadie, D. C. Garcia, and K. D. Luks, Fluid Phase Equilibria, **171**, 11–26 (2000).

[16] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*. (Clarendon Press, Oxford, 1987).

[17] A. Z. Panagiotopoulos, Molecular Physics, **61**(4), 813–826 (1987).

[18] A. Z. Panagiotopoulos, N. Quirke, M. Stapleton, and D. J. Tildesley, Molecular Physics, **63**(4), 527–545 (1988).

[19] D. A. Kofke, Molecular Physics, **78**(6), 1331–1336 (1993).

[20] D. A. Kofke, Journal of Chemical Physics, **98**(5), 4149–4162 (1993).

[21] A. M. Georgolaki, I. V. Ntouros, D. P. Tassios, and A. Z. Panagiotopoulos, Fluid Phase Equilibria, **100**, 153–170 (1994).

[22] V. I. Harismiadis, N. K. Kourtras, D. P. Tassios, and A. Z. Panagiotopoulos, Fluid Phase Equilibria, **65**, 1–18 (1991).

[23] P. C. Tsang, O. N. White, B. Y. Perigard, L. F. Vega, and A. Z. Panagiotopoulos, Fluid Phase Equilibria, **107**, 31–43 (1995).

[24] J. N. C. Lopes, Molecular Physics, **96**(11), 1649–1658 (1999).

[25] J. N. C. Lopes and D. J. Tildesley, Molecular Physics, **92**(3), 187–195 (1997).

[26] M. Guo, Y. Li, Z. Li, and J. Lu, Fluid Phase Equilibria, **98**, 129–139 (1994).

[27] M. E. van Leeuwen, C. J. Peters, J. de Swaan Arons, and A. Z. Panagiotopoulos, Fluid Phase Equilibria, **66**, 57–75 (1991).

[28] R. J. Sadus, Fluid Phase Equilibria, **157**, 169–180 (1999).

[29] M. R. Hitchcock and C. K. Hall, Journal of Chemical Physics, **110**(23), 11433–11444 (1999).

[30] J. S. Rowlinson and F. L. Swinton, *Liquids and Liquid Mixtures*. (Butterworth Scientific, London, 1982).

[31] M. R. Hitchcock and C. K. Hall. *Complete Phase Diagrams for Binary Mixtures via Gibbs-Duhem Integration*, (2000). To appear in Proceedings of the Foundations of Molecular Modeling and Simulation Conference.

[32] M. H. Lamm and C. K. Hall. *Monte Carlo Simulations of Complete Phase Diagrams for Binary Lennard-Jones Mixtures*, (2000). Fluid Phase Equilibria, in press.

[33] M. J. Vlot, J. C. van Miltenburg, H. A. J. Oonk, and J. P. van der Eerden, Journal of Chemical Physics, **107**(23), 10102–10111 (1997).

[34] R. Agrawal and D. A. Kofke, Molecular Physics, **85**(1), 43–59 (1995).

[35] D. A. Kofke, in *Monte Carlo Methods in Chemistry* edited by D. M. FergusonJ. I. Siepmann and D. G. Truhlar (Interscience Publishers, New York, 1998).

[36] M. Mehta and D. A. Kofke, Chemical Engineering Science, **49**(16), 2633–2645 (1994).

[37] B. Carnahan, H. A. Luther, and J. O. Wilkes, *Applied Numerical Methods*. (John Wiley & Sons, New York, 1969).

[38] P. Gordon, *Principles of Phase Diagrams in Materials Systems*. (Krieger, Malabar, Florida, 1983).

[39] A. Prince, *Alloy Phase Equilibria*. (Elsevier, Amsterdam, 1966).

[40] D. A. Kofke and E. D. Glandt, Molecular Physics, **64**(6), 1105–1131 (1988).

[41] J.-P. Hansen and L. Verlet, Physical Review, **184**, 151–161 (1969).

[42] D. A. Kofke, Molecular Simulation, **7**, 285–302 (1991).

[43] W. Hume-Rothery, R. E. Smallman, and C. W. Haworth, *The Structure of Metals and Alloys*. (The Metals and Metallurgy Trust, London, 1969).

2.6 Figures

Page

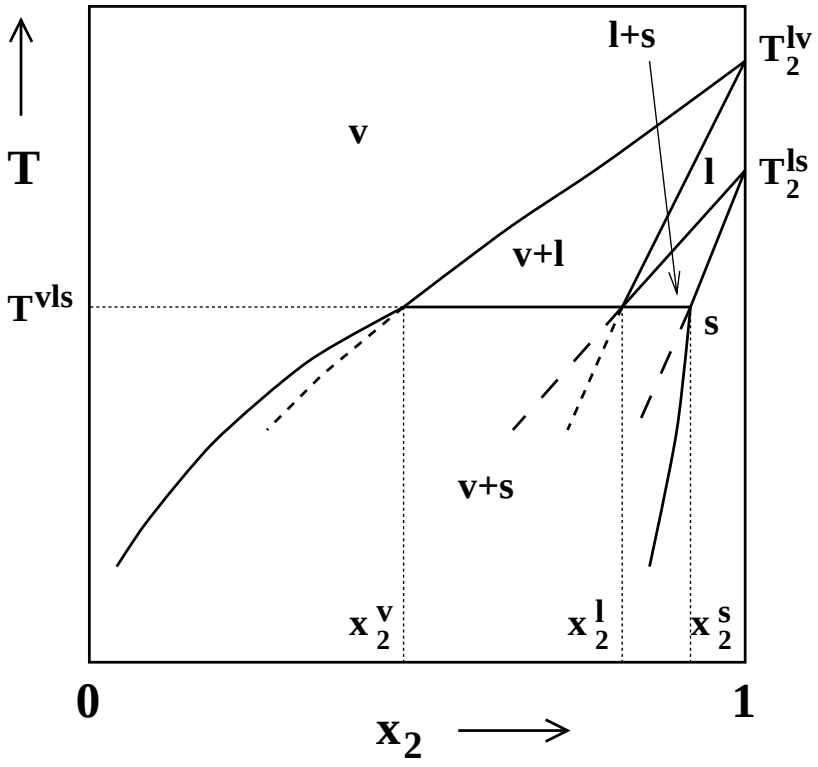

Figure 2.1: Illustration of the procedure used to determine a solid-liquid-vapor coexistence line (see text for description). Solid lines show the equilibrium phase boundaries on the $T - x_2$ diagram. The labels v (vapor), l (liquid), and s (solid) identify the phases present in each region. Short-dashed and long-dashed lines represent the metastable parts of the vapor-liquid and solid-liquid coexistence curves, respectively. Other labels are as follows: T_2^{lv}, vapor-liquid coexistence temperature for component 2; T_2^{ls}, solid-liquid coexistence temperature for component 2, T^{vls}, vapor-liquid-solid coexistence temperature; x_2^v, x_2^l, x_2^s, the vapor, liquid, and solid phase coexistence mole fractions for component 2.

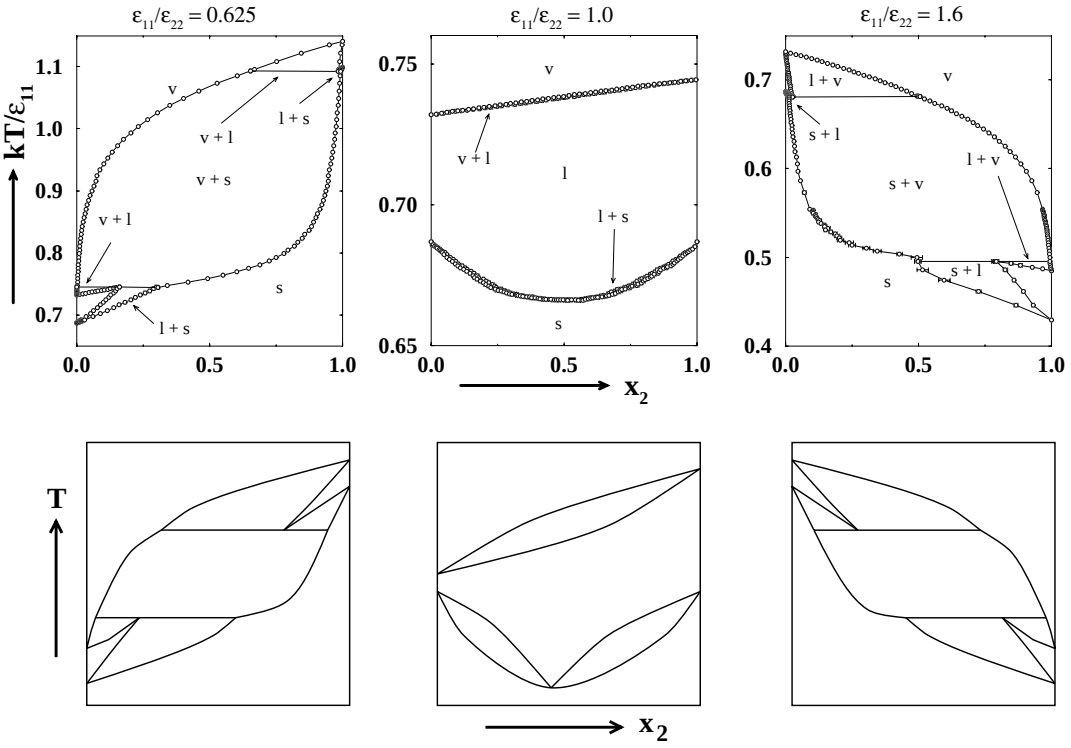

Figure 2.2: Temperature vs. composition phase diagrams for Lennard-Jones binary mixtures with diameter ratio $\sigma_{11}/\sigma_{22} = 0.95$ at $P^* = 0.002$. Row 1 shows the phase diagrams obtained via Gibbs-Duhem integration. The circles represent data from Gibbs-Duhem integration simulations. Error bars are shown when they are larger than the width of the symbol. Lines are drawn through the points for clarity. The labels identifying the phases present in each region are as follows: v (vapor), l (liquid), and s (solid solution). Row 2 gives schematic diagrams (not to scale) corresponding to the phase diagrams in row 1.

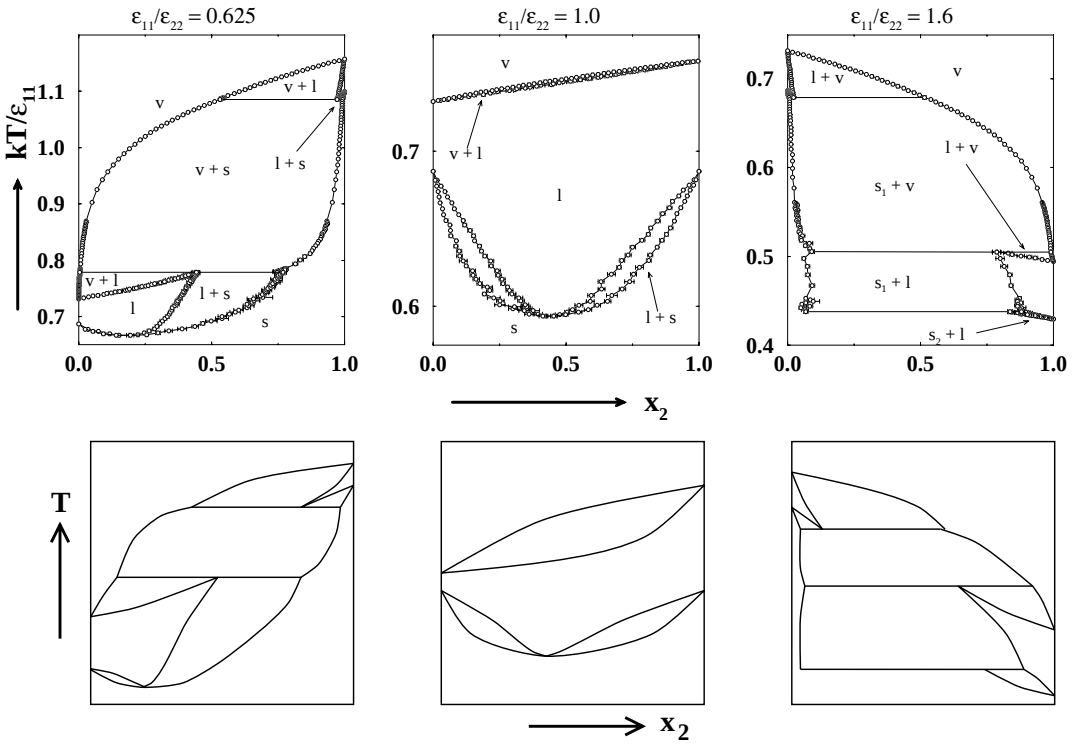

Figure 2.3: Temperature vs. composition phase diagrams for Lennard-Jones binary mixtures with diameter ratio $\sigma_{11}/\sigma_{22} = 0.90$ at $P^* = 0.002$. Row 1 shows the phase diagrams obtained via Gibbs-Duhem integration. The labels s_1 and s_2 denote solid solutions rich in component 1 and 2, respectively). Other symbols as in Figure 2.2. Row 2 gives schematic diagrams (not to scale) corresponding to the phase diagrams in row 1.

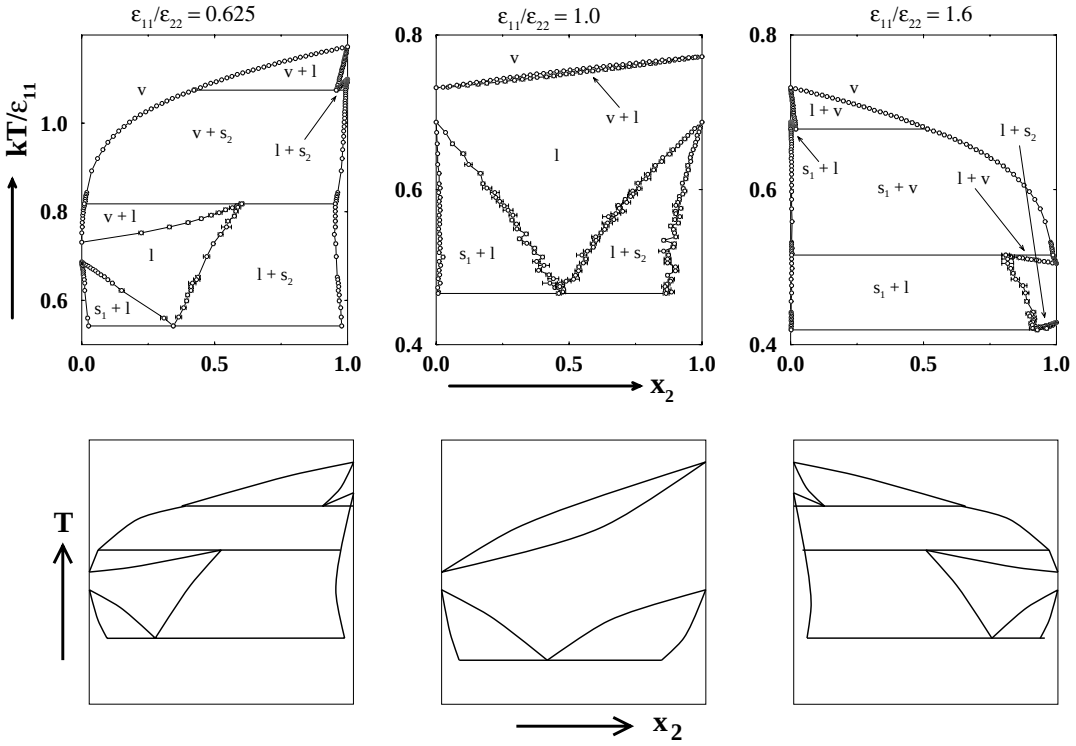

Figure 2.4: Temperature vs. composition phase diagrams for Lennard-Jones binary mixtures with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$ at $P^* = 0.002$. Row 1 shows the phase diagrams obtained via Gibbs-Duhem integration. The labels s_1 and s_2 denote solid solutions rich in component 1 and 2, respectively). Other symbols as in Figure 2.2. Row 2 gives schematic diagrams (not to scale) corresponding to the phase diagrams in row 1.

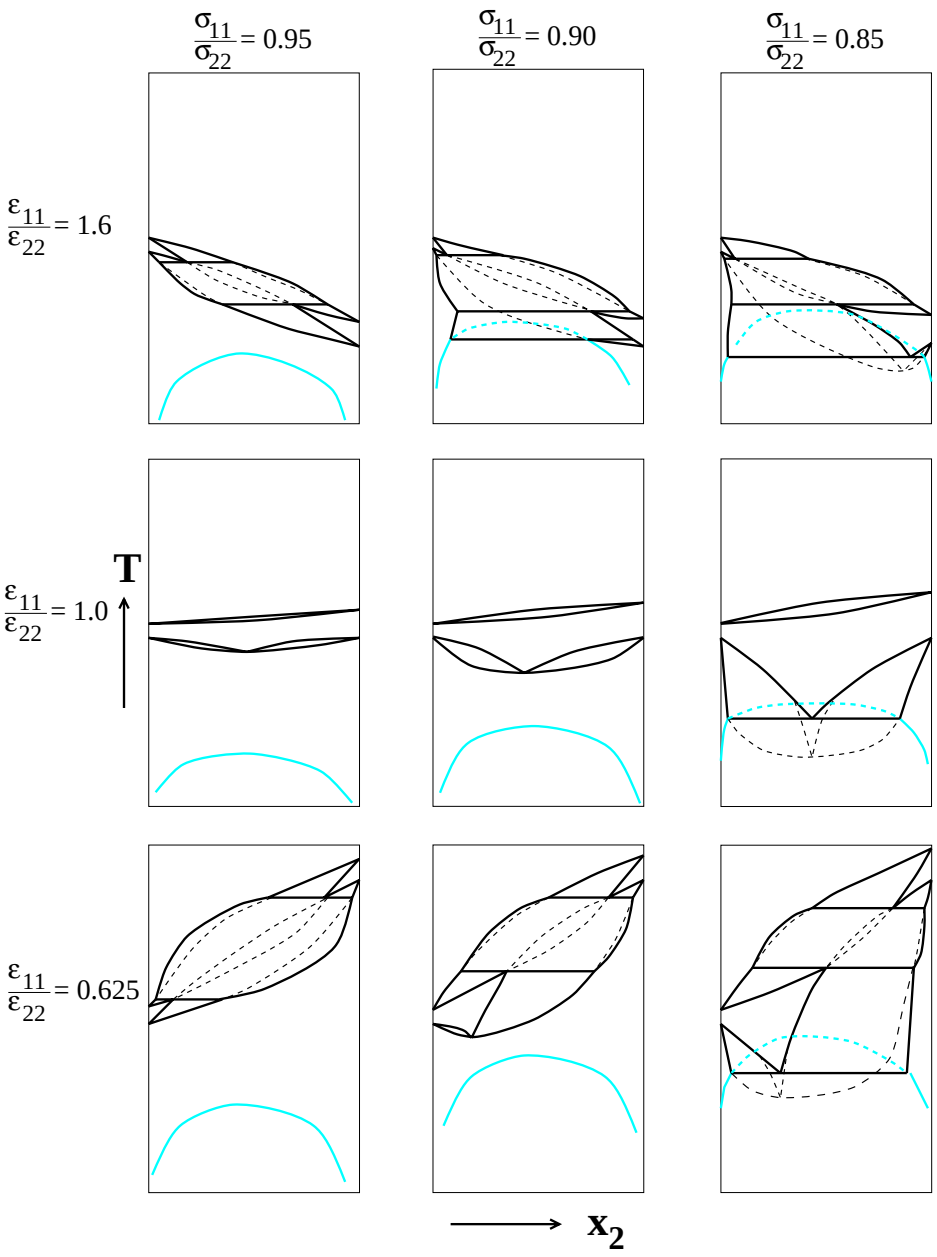

Figure 2.5: Schematic diagrams illustrating the interference between vapor-liquid, solid-liquid, and solid-solid coexistence regions. The columns correspond to mixtures with diameter ratios σ_{11}/σ_{22} = 0.95, 0.9, and 0.85, and the rows correspond to mixtures with well-depth ratios $\epsilon_{11}/\epsilon_{22}$ = 1.6, 1.0, and 0.625. Solid black lines correspond to equilibrium curves calculated in this work. Solid gray lines represent solid-solid coexistence curves, which have not been calculated for these mixtures but have been estimated using quasichemical theory. Metastable coexistence curves are indicated with dashed lines.

CHAPTER 3

THE EFFECT OF THE BINARY INTERACTION PARAMETER ON COMPLETE PHASE BEHAVIOR OF BINARY MIXTURES

3.1 Introduction

Phase equilibria for binary mixtures of spherically symmetric molecules has been the subject of intensive investigation for decades. Van Konynenburg and Scott[1,2] classified the phase diagrams predicted by the van der Waals equation of state for binary mixtures. Remarkably, the simple van der Waals equation of state was found to exhibit five of the six types of fluid phase behavior observed experimentally. This landmark study has been followed by similar analyses for other equations of state, such as the Redlich-Kwong[3], the Carnahan-Starling-Redlich-Kwong[4], the Guggenheim[5], and the Ree[6] equations of state.

Most of the research directed at understanding how intermolecular interactions affect phase behavior has focused exclusively on fluid phase equilibria. However in real systems solid phases form and often interrupt the complex fluid phase be-

havior[7,8]. Phenomenological descriptions of complete phase diagrams (*i.e.*, showing equilibrium between vapor, liquid, and solid phases) have been given by Luks[9], Valyashko[10,11], and Peters, Lichtenthaler and de Swaan Arons[12]. Valyashko proposed a classification scheme for complete diagrams and introduced new types of complete phase behavior that have not yet been observed in real systems. Garcia and Luks[13] recently examined the solid-liquid-vapor locus for a series of binary mixtures using the van der Waals equation of state and a simple solid state fugacity model. In their model calculations, they found examples of solid-fluid phase behavior in keeping with what has been observed in real systems, as well as solid-fluid phase behavior that has yet to be verified by experiment. The new possibilities for complete phase behavior proposed by Valyashko and by Garcia and Luks are intriguing and invite further investigation.

Molecular simulation has become a popular way to study the phase behavior of model systems[14]. Numerous simulations of binary mixture phase behavior for the Lennard-Jones intermolecular potential, the quintessential model of a spherically symmetric molecule, have been conducted for both fluid-fluid[15–21] and solid-liquid[22] phase equilibria. Vlot, van Miltenburg, Oonk, and van der Eerden[23] calculated complete phase diagrams for symmetric (equal diameters, $\sigma_{11} = \sigma_{22}$; equal attractions $\epsilon_{11} = \epsilon_{22}$) Lennard-Jones mixtures by using Monte Carlo simulation for selected state points to determine the excess free energy as a function of composition. This free energy versus composition data was fit with a two parameter Redlich-Kister polynomial and the convex envelope construction method was used to determine the

phase diagram.

The Lennard-Jones intermolecular potential is given by

$$u_{ij}(r) = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r}\right)^{12} - \left(\frac{\sigma_{ij}}{r}\right)^{6}\right], \tag{3.1}$$

where u_{ij} is the potential energy of interaction between particles i and j, r is the distance between particles i and j, ϵ_{ij} is the Lennard-Jones attractive well-depth, and σ_{ij} is the Lennard-Jones diameter. The cross-species interaction parameters $(\sigma_{12}, \epsilon_{12})$ are determined by the Lorentz-Berthelot combining rules[24]

$$\sigma_{12} = (\sigma_{11} + \sigma_{22})/2$$

$$\epsilon_{12} = \delta_{12}\sqrt{\epsilon_{11}\epsilon_{22}}$$

where δ_{12}, the binary interaction parameter, accounts for deviations of the unlike-pair attraction from the geometric mean of the like-pair attractions. According to the global phase diagrams associated with the various equations of state[2–6] for binary mixtures of equal-size, spherically symmetric molecules, liquid-liquid immiscibility occurs whenever the unlike-pair attractions are less than the arithmetic mean of the like-pair attractions. This means that liquid-liquid immiscibility is predicted to occur when

$$\delta_{12} < \frac{\epsilon_{11} + \epsilon_{22}}{2\sqrt{\epsilon_{11}\epsilon_{22}}}. \tag{3.2}$$

For binary mixtures of unequal-size, spherically symmetric molecules, liquid-liquid

immiscibility is predicted to occur when

$$\delta_{12} < \frac{\epsilon_{11} + \epsilon_{22}(\sigma_{11}/\sigma_{22})^3}{2\sqrt{\epsilon_{11}\epsilon_{22}}(\sigma_{11}/\sigma_{22}+1)^3}. \tag{3.3}$$

Liquid-liquid immiscibility might not be observed however, because of intervening solid phases, which cannot be predicted by the equations of state. In fact, Rowlinson and Swinton[24] point out that unless the unlike-pair attractions are significantly weaker than the like-pair attractions, the liquid-liquid upper critical solution temperature resides at temperatures below the quadruple point ($s_1 s_2 l v$).

Systematic studies of the effect of δ_{12} on fluid phase behavior have been conducted for two Lennard-Jones binary mixtures[15,17] using Gibbs ensemble simulation. In both cases, δ_{12} was varied over the range 1.0 to 0.7 to observe the effect of the unlike-pair attractions on the constant temperature P-x phase diagram. In the first mixture ($\sigma_{11}/\sigma_{22} = 0.82$, $\epsilon_{11}/\epsilon_{22} = 0.52$), Panagiotopoulos[15] found that liquid-liquid immiscibility occurred for $\delta_{12} \leq 0.8$ at $kT/\epsilon_{11} = 1.348$. In the second mixture ($\sigma_{11}/\sigma_{22} = 0.94$, $\epsilon_{11}/\epsilon_{22} = 0.73$), van Leeuwen, Peters, de Swaan Arons, and Panagiotopoulos[17] found that liquid-liquid immiscibility occurred for $\delta_{12} \leq 0.75$ at $kT/\epsilon_{11} = 1.218$. A possible explanation for why liquid-liquid immiscibility was not observed for $\delta_{12} = 1.0$ in these two studies is that the simulations were conducted at fixed temperatures that were higher than the upper critical solution temperature.

In this chapter we calculate complete phase diagrams for binary Lennard-Jones mixtures using Gibbs-Duhem integration combined with semigrand canonical Monte Carlo simulation. Our objective is to explore the effect of δ_{12} on the phase behavior

of a mixture when solid phase formation is included in the calculation. We present complete phase diagrams for binary Lennard-Jones mixtures with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$, well-depth ratio $\epsilon_{11}/\epsilon_{22} = 0.45$, and binary interaction parameters $\delta_{12} = 1.0, 0.9,$ and 0.75. at reduced pressure, $P^* = P\sigma_{11}^3/\epsilon_{11} = 0.05$.

The remainder of this chapter is organized as follows. We first outline the Gibbs-Duhem integration method and describe how we applied the procedure to the calculation of complete phase behavior. We then present the resulting complete phase diagrams.

3.2 Method

3.2.1 Gibbs-Duhem integration

The coexistence lines were calculated using Gibbs-Duhem integration[25–27]. In this method, phase coexistence is determined by numerically integrating the Clapeyron differential equation appropriate to the system of interest. Clapeyron equations describe how field variables (variables that must be equal among coexisting phases) change along the phase equilibrium line. The Clapeyron equation for equilibrium between two phases (α and γ) of a binary mixture containing components 1 and 2 at constant pressure is

$$\frac{d\beta}{d\xi_2} = \frac{(x_2^\alpha - x_2^\gamma)}{\xi_2(1-\xi_2)(h^\alpha - h^\gamma)}, \tag{3.4}$$

where β is the reciprocal temperature, $1/kT$, with k the Boltzmann constant and T the absolute temperature, ξ_2 is the fugacity fraction of species 2, $\xi_2 \equiv \hat{f}_2/\sum \hat{f}_i$, with

$\hat{f}_i$, the fugacity of species i in solution, x_2 is the mole fraction of species 2, and h is the molar enthalpy. The right-hand side of Eq. (3.4) can be integrated numerically to find an equation for β as a function of ξ_2 if we have an initial condition describing the temperature, fugacity fraction, enthalpies and compositions at one coexistence point.

A convenient choice for the initial coexistence condition is the vapor-liquid or solid-liquid equilibrium condition for either of the pure components. Here, we used single component Lennard-Jones vapor-liquid[26], solid-liquid[28] coexistence data taken from the literature. The slope of the integrand in Eq. (3.4) is undefined for pure components ($\xi_2 = 0, x_2 = 0$ and $\xi_2 = 1, x_2 = 1$) but it can be estimated using the limiting case of infinite dilution. This procedure is described in Chapter 2 and will not be repeated here.

Once we have an initial coexistence condition, we begin the Gibbs-Duhem integration procedure using a predictor-corrector algorithm to integrate Eq. (3.4) over the entire range of fugacity fractions, $\xi_2 = 0$ to $\xi_2 = 1$. The algorithm proceeds as follows. A step in fugacity fraction, $\Delta\xi_2$, is made and the next reciprocal temperature, β, is estimated using the trapezoid-rule predictor formula. After β is estimated, two semigrand canonical (NPTξ_2) Monte Carlo simulations (one for the α phase and one for the γ phase) are conducted in order to equilibrate the configurations and to calculate the enthalpies and mole fractions at the new state point. (Details of the NPTξ_2 simulations are given below.) After each phase is equilibrated the estimate is refined with the trapezoid-rule corrector. After β is corrected, two NPTξ_2 simu-

lations, one for each phase, are conducted to obtain averages of the enthalpies and mole fractions. Each simulation is paused periodically to correct the β estimate, until β stops varying within a specified tolerance. A final production segment of simulations are run at the temperature determined from the corrector segment to obtain the final average enthalpies and mole fractions for the coexistence point. Once the production runs are completed, the fugacity fraction is incremented to the next state point and the predictor-corrector algorithm described above is repeated. Higher order predictor-corrector equations are used as more coexistence points are obtained. An entire temperature versus composition coexistence curve can be mapped out by repeating the predictor-corrector algorithm from $\xi_2 = 0$ to $\xi_2 = 1$.

In some of the mixtures, we encountered occurrences of three-phase coexistence lines, such as solid-liquid-vapor equilibrium. In this case, two Gibbs-Duhem integrations were conducted, the first starting from the vapor-liquid coexistence temperature for one of the pure components and the second starting from the solid-liquid coexistence temperature for the same pure component. For example, in Figure 3.1 vapor-liquid and solid-liquid coexistence curves have been calculated starting from the vapor-liquid and solid-liquid coexistence temperatures (T_2^{lv} and T_2^{sl}, respectively) of pure component 2. At some ξ_2 (unknown at the commencement of the two integrations) the liquid phase coexistence lines from each set of coexistence curves will cross, thus, determining the temperature, fugacity fraction, and coexistence compositions of the three coexisting phases: vapor, liquid, and solid. In Figure 3.1 the liquid lines have crossed at T^{vls} and x_2^l. Below T^{vls} the vapor-liquid

and solid-liquid coexistence curves (shown by short-dashed and long-dashed lines, respectively) are metastable with respect to vapor-solid equilibrium. The vapor-solid coexistence curve is determined by a Gibbs-Duhem integration starting from T^{vls}, x_2^v, and x_2^s. Other types of three-phase coexistence lines (heteroazeotropes, eutectics, etc.) can be determined in a similar manner.

3.2.2 Simulations

The enthalpies and mole fractions needed as input to the integration of Eq. (3.4) are obtained from semigrand canonical (constant NPTξ_2) Monte Carlo computer simulations[30]. In this work, all simulations were run with a system size of 500 particles at a reduced pressure $P^* = 0.05$. The temperature and fugacity fraction were varied according to the values specified by the Gibbs-Duhem integration predictor-corrector. There are three types of Monte Carlo trial moves in semigrand canonical simulations: particle displacements, volume change moves, and particle identity exchanges. The particle displacements and volume change moves are conducted just as they are in a standard NPT simulation[31]. In the particle identity exchange moves, a particle is selected at random and given a trial species identity switch, *i.e.*, if a species 1 particle is chosen it is given a trial identity of species 2 and vice versa. The identity exchange move is accepted according to the ratio of the species fugacity fractions, ξ_1 and ξ_2. The overall acceptance probability[29] for the moves in the

NPTξ_2 ensemble is $\min[1, \exp(\Lambda)]$ where

$$\Lambda = -\beta(U^{\text{trial}} - U^{\text{old}}) - \beta P(V^{\text{trial}} - V^{\text{old}}) + N \ln \frac{V^{\text{trial}}}{V^{\text{old}}} + m \ln \frac{\xi_2}{1 - \xi_2}. \tag{3.5}$$

In Eq. (3.5), U^{trial} and U^{old}, and V^{trial} and V^{old}, are the configurational energies and volumes of the trial and existing states, respectively, $m = +1$ if the trial identity exchange is from species 1 to 2, and $m = -1$ if the trial identity exchange is from species 2 to 1. The length of the simulation is given in cycles, where one cycle represents 1 volume change attempt, N particle displacement attempts,and N particle identity exchange attempts. In our work, a typical NPTξ_2 simulation is equilibrated for 3000 cycles and then followed by a production run of 5000 cycles to compute the average enthalpy and mole fraction.

Other details of the NPTξ_2 simulations are as follows. The simulation volume is a cubic box with periodic boundary conditions. The potential interactions are truncated at a cutoff radius of half the box length. To compensate for this truncation, a long range correction is applied to the energy and virial calculations during the simulation by assuming a uniform density distribution beyond the cutoff radius[31]. To maintain an fcc crystalline structure in the solid phase simulations, we impose a single occupancy constraint[32,33] on the trial displacements of particles in the solid, *i.e.,* any displacements that put the particle outside its lattice cell are rejected.

3.3 Results

In this section, we present the results of our Gibbs-Duhem integration calculations of complete phase behavior for binary Lennard-Jones mixtures.

We calculated three complete phase diagrams at $P^* = P\sigma_{11}^3/\epsilon_{11} = 0.05$ for diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$, well-depth ratio $\epsilon_{11}/\epsilon_{22} = 0.45$, and binary interaction parameters $\delta_{12} = 1.0, 0.9,$ and 0.75. According to Eq. (3.3) (based on equation of state predictions), liquid-liquid immiscibility is predicted to occur for this mixture whenever $\delta_{12} < 1.002$. Figure 3.2 shows a temperature-composition phase diagram for $\delta_{12} = 1.0$. The phase diagram starts from the vapor-liquid coexistence temperature for pure component 2 ($x_2 = 1, T^* = 2.368$). The vapor-liquid coexistence lines decrease in temperature with decreasing fugacity fraction until they reach the vapor-liquid coexistence temperature for pure component 1 ($x_2 = 0, T^* = 1.121$). A miscible liquid phase exists below the vapor-liquid curves. Solid-liquid coexistence lines originate from both pure component 1 ($x_2 = 0, T^* = 0.691$) and pure component 2 ($x_2 = 1, T^* = 1.533$) and decrease in temperature with varying fugacity fraction until they meet to form a eutectic at $T^* = 0.652$. At this temperature three phases coexist: a solid solution rich in component 1 (solid(1)), a liquid mixture, and solid solution rich in component 2 (solid(2)). Below this temperature solid(1) and solid(2) are in equilibrium (solid-solid coexistence lines were not calculated). No evidence for liquid-liquid equilibria was observed over temperature range, $T^* = 0.652 - 2.638$, as one would expect since the calculations took place above the quadruple point.

Figure 3.3a shows a temperature-composition phase diagram for $\delta_{12} = 0.9$.

Vapor-liquid coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 2.368$) and decrease in temperature with decreasing fugacity fraction. Liquid-solid(2) coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 1.533$) and decrease in temperature with decreasing fugacity fraction. The vapor-liquid and liquid-solid(2) curves meet at $T^* = 1.457$ and form a three-phase, vapor-liquid-solid(2) equilibrium line. Vapor-solid(2) coexistence lines originate from this three-phase line and decrease in temperature with decreasing fugacity fraction. Vapor-liquid coexistence lines originate from pure component 1 ($x_2 = 0, T^* = 1.121$) and increase in temperature with increasing fugacity fraction. The vapor-liquid and vapor-solid(2) lines meet at $T^* = 1.130$ to form another three-phase, vapor-liquid-solid(2) equilibrium line. Figure 3.3b shows an enlargement of the phase diagram near the intersection of the vapor-liquid and vapor-solid(2) regions. Liquid-solid(2) coexistence lines originate from this three-phase line and decrease in temperature with decreasing fugacity fraction. Solid(1)-liquid coexistence lines originate from pure component 1 ($x_2 = 0, T^* = 0.691$) and decrease in temperature with increasing fugacity fraction. The solid(1)-liquid and liquid-solid(2) coexistence curves meet at $T^* = 0.690$ to form a eutectic. Figure 3.3c shows an enlargement of the phase diagram near the intersection of the solid(1)-liquid and liquid-solid(2) regions. Below this temperature solid(1) and solid(2) are in equilibrium.

In this mixture, the solid phase interfered before we could determine whether the liquid phase would separate. If any liquid-liquid immiscibility exists, it would be metastable with respect to the solid-fluid phase equilibria. Since the Gibbs-Duhem

integration method can be used to calculate metastable coexistence lines in addition to equilibrium coexistence lines, we continued our earlier calculations of the vapor-liquid coexistence curves as if there were no solid phase interference. Figure 3.4a shows the resulting metastable coexistence lines. Vapor-liquid coexistence curves continue from $T^* = 1.457$ and decrease in temperature with decreasing fugacity fraction. Similarly, vapor-liquid coexistence curves continue from $T^* = 1.13$ and increase in temperature with increasing fugacity fraction. At $T^* = 1.137$, the two sets of vapor-liquid coexistence curves meet and form a heteroazeotrope. At this temperature three phases coexist: a vapor, a liquid mixture rich in component 1 (liquid(1)) and a liquid mixture rich in component 2 (liquid(2)). Liquid(1)-liquid(2) coexistence lines originate from this heteroazeotrope and decrease in temperature with decreasing fugacity fraction. Figure 3.4b is a schematic phase diagram to illustrate more clearly the features of the phase diagram shown in Figure 3.4a.

Figure 3.5 shows a temperature-composition phase diagram for $\delta_{12} = 0.75$. This phase diagram has the same qualitative features as the phase diagram shown for $\delta_{12} = 0.9$ in Figure 3.3. Vapor-liquid-solid(2) coexistence occurs at $T^* = 1.515$ and $T^* = 1.122$ and a eutectic occurs at $T^* = 0.6908$. Metastable liquid-liquid immiscibility (not shown in Figure 3.5) was also found for this mixture, with the metastable heteroazeotrope occurring at $T^* = 1.122$.

3.4 Summary

The Gibbs-Duhem integration technique was combined with semigrand canonical Monte Carlo simulations to calculate complete $T - x$ phase diagrams for binary Lennard-Jones mixtures. We studied the effect of unlike-pair attractions on the mixture phase diagram by varying the binary interaction parameter over the range $\delta_{12} =$ 1 to 0.75 while holding the diameter ratio and well-depth ratio constant.

For the mixture with $\delta_{12} = 1$ we found a completely miscible vapor-liquid coexistence region and a eutectic solid-liquid coexistence region. These two regions were separated by a completely miscible liquid phase. No evidence for liquid-liquid phase separation was found over the temperature range, $T^* = 0.652$ to 2.638. Limitations in the available methods for simulating phase coexistence (Gibbs-Duhem integration and Gibbs ensemble Monte Carlo) prevent us from searching for liquid-liquid immiscibility below $T^* = 0.652$. Gibbs-Duhem integration simulations cannot be conducted without first having an initial liquid-liquid coexistence condition. Gibbs ensemble simulations are ineffective below the triple point temperature of the pure Lennard-Jones fluid ($T^* = 0.687$) because the densities of the coexisting phases are too high to permit successful particle transfers between phases[15]. We cannot, therefore, rule out the possibility of observing a metastable upper critical solution temperature at a lower temperature.

For the mixtures with $\delta_{12} = 0.9$ and 0.75 we found that the vapor-liquid and solid-liquid coexistence regions interfered. This interference resulted in a vapor-solid coexistence region bounded above and below by solid-liquid-vapor coexistence

lines. We also found that these two mixtures had a region of liquid-liquid immiscibility that was metastable with respect to the solid-fluid phase equilibria. Unlike the case for $\delta_{12} = 1$, we were able to calculate the metastable liquid-liquid coexistence lines with Gibbs-Duhem integration because they originated from a three-phase equilibrium line for which we had coexistence data.

3.5 References

[1] R. L. Scott and P. H. van Konynenburg, Discussions of the Faraday Society, **49**, 87–97 (1970).

[2] P. H. van Konynenburg and R. L. Scott, Philosophical Transactions of the Royal Society of London A., **298**, 495–540 (1980).

[3] U. K. Deiters and I. L. Pegg, Journal of Chemical Physics, **90**, 6632–6641 (1989).

[4] T. Kraska and U. K. Deiters, Journal of Chemical Physics, **96**, 539–547 (1992).

[5] J. Wang, G. W. Wu, and R. J. Sadus, Molecular Physics, **98**, 715–723 (2000).

[6] V. A. Mazur, L. Z. Boshkov, and V. G. Murakhovsky, Physics Letters, **104**, 415–417 (1984).

[7] G. M. Schneider, in *Chemical Thermodynamics* edited by M. L. McGlashan (The Chemical Society, London, 1978).

[8] R. L. Scott, Accounts of Chemical Research, **20**, 97–107 (1987).

[9] K. D. Luks, Fluid Phase Equilibria, **29**, 209–224 (1986).

[10] V. M. Valyashko, Z. Phys. Chemie, Leipzig, **267**(3), 481–493 (1986).

[11] V. M. Valyashko, Pure and Applied Chemistry, **62**(11), 2129–2138 (1990).

[12] C. J. Peters, R. N. Lichtenthaler, and J. de Swaan Arons, Fluid Phase Equilibria, **29**, 495–504 (1986).

[13] D. C. Garcia and K. D. Luks, Fluid Phase Equilibria, **161**, 91–106 (1999).

[14] A. Z. Panagiotopoulos, Journal of Physics: Condensed Matter, **12**, R25–R52 (2000).

[15] A. Z. Panagiotopoulos, in *Supercritical Fluid Science and Technology* edited by K. P. Johnston and J. Penninger (American Chemical Society, 1989).

[16] V. I. Harismiadis, N. K. Kourtras, D. P. Tassios, and A. Z. Panagiotopoulos, Fluid Phase Equilibria, **65**, 1–18 (1991).

[17] M. E. van Leeuwen, C. J. Peters, J. de Swaan Arons, and A. Z. Panagiotopoulos, Fluid Phase Equilibria, **66**, 57–75 (1991).

[18] A. M. Georgolaki, I. V. Ntouros, D. P. Tassios, and A. Z. Panagiotopoulos, Fluid Phase Equilibria, **100**, 153–170 (1994).

[19] M. Guo, Y. Li, Z. Li, and J. Lu, Fluid Phase Equilibria, **98**, 129–139 (1994).

[20] J. N. C. Lopes, Molecular Physics, **96**(11), 1649–1658 (1999).

[21] R. J. Sadus, Fluid Phase Equilibria, **157**, 169–180 (1999).

[22] M. R. Hitchcock and C. K. Hall, Journal of Chemical Physics, **110**(23), 11433–11444 (1999).

[23] M. J. Vlot, J. C. van Miltenburg, H. A. J. Oonk, and J. P. van der Eerden, Journal of Chemical Physics, **107**(23), 10102–10111 (1997).

[24] J. S. Rowlinson and F. L. Swinton, *Liquids and Liquid Mixtures*. (Butterworth Scientific, London, 1982).

[25] D. A. Kofke, Molecular Physics, **78**(6), 1331–1336 (1993).

[26] D. A. Kofke, Journal of Chemical Physics, **98**(5), 4149–4162 (1993).

[27] D. A. Kofke, in *Monte Carlo Methods in Chemistry* edited by D. M. FergusonJ. I. Siepmann and D. G. Truhlar (Interscience Publishers, New York, 1998).

[28] R. Agrawal and D. A. Kofke, Molecular Physics, **85**(1), 43–59 (1995).

[29] M. Mehta and D. A. Kofke, Chemical Engineering Science, **49**(16), 2633–2645 (1994).

[30] D. A. Kofke and E. D. Glandt, Molecular Physics, **64**(6), 1105–1131 (1988).

[31] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*. (Clarendon Press, Oxford, 1987).

[32] J.-P. Hansen and L. Verlet, Physical Review, **184**, 151–161 (1969).

[33] D. A. Kofke, Molecular Simulation, **7**, 285–302 (1991).

3.6 Figures

Page

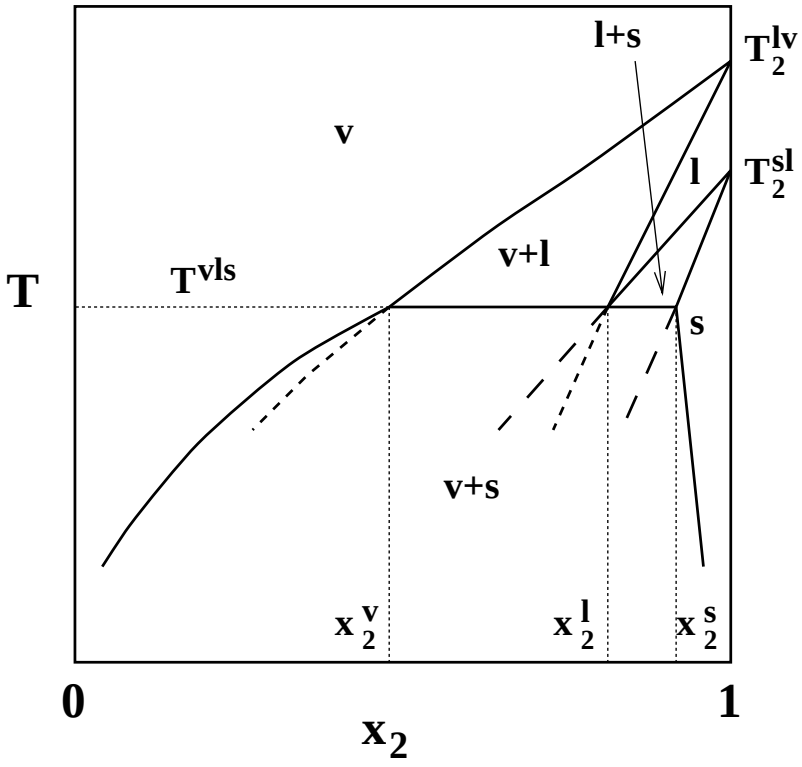

Figure 3.1: Illustration of the procedure used to determine a solid-liquid-vapor coexistence line (see text for description). Solid lines show the phase boundaries on the $T - x_2$ diagram. The labels v (vapor), l (liquid), and s (solid) identify the phases present in each region. Short-dashed and long-dashed lines represent the metastable parts of the vapor-liquid and solid-liquid coexistence curves, respectively. Other labels are as follows: T_2^{lv}, vapor-liquid coexistence temperature for component 2; T_2^{sl}, solid-liquid coexistence temperature for component 2, T^{vls}, vapor-liquid-solid coexistence temperature; x_2^v, x_2^l, x_2^s, the vapor, liquid, and solid phase coexistence mole fractions for component 2.

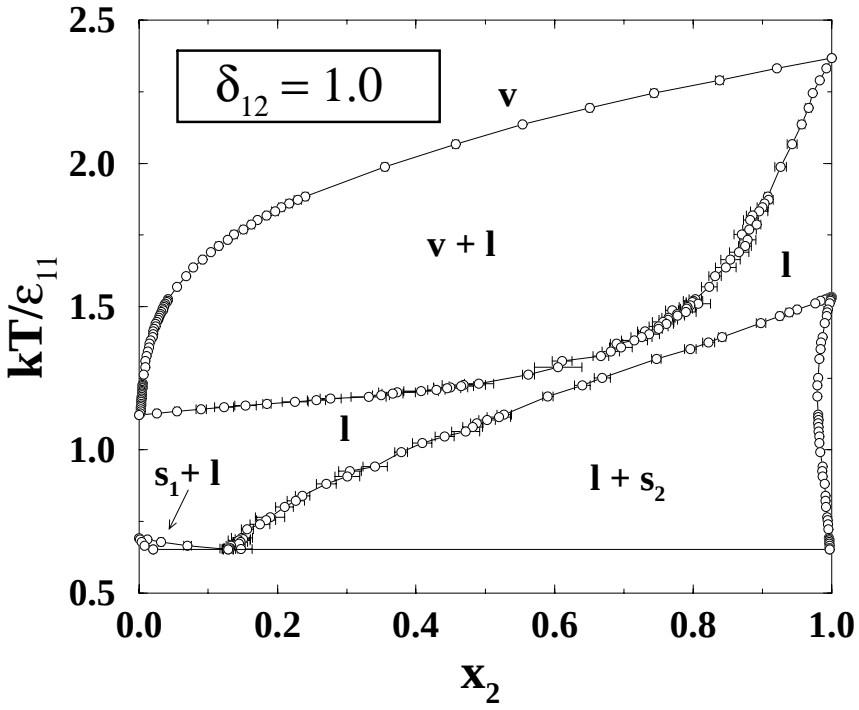

Figure 3.2: Complete temperature vs. composition phase diagram at $P^* = 0.05$ for a binary Lennard-Jones mixture with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$, well-depth ratio $\epsilon_{11}/\epsilon_{22} = 0.45$, and binary interaction parameter $\delta_{12} = 1.0$. The circles represent data from Gibbs-Duhem integration simulations. Error bars are shown when they are larger that the width of the symbol. Lines are drawn through the points for clarity. The labels identifying the phases present in each region are as follows: v (vapor), l (liquid), s_1, and s_2 (solid solutions rich in components 1 and 2, respectively).

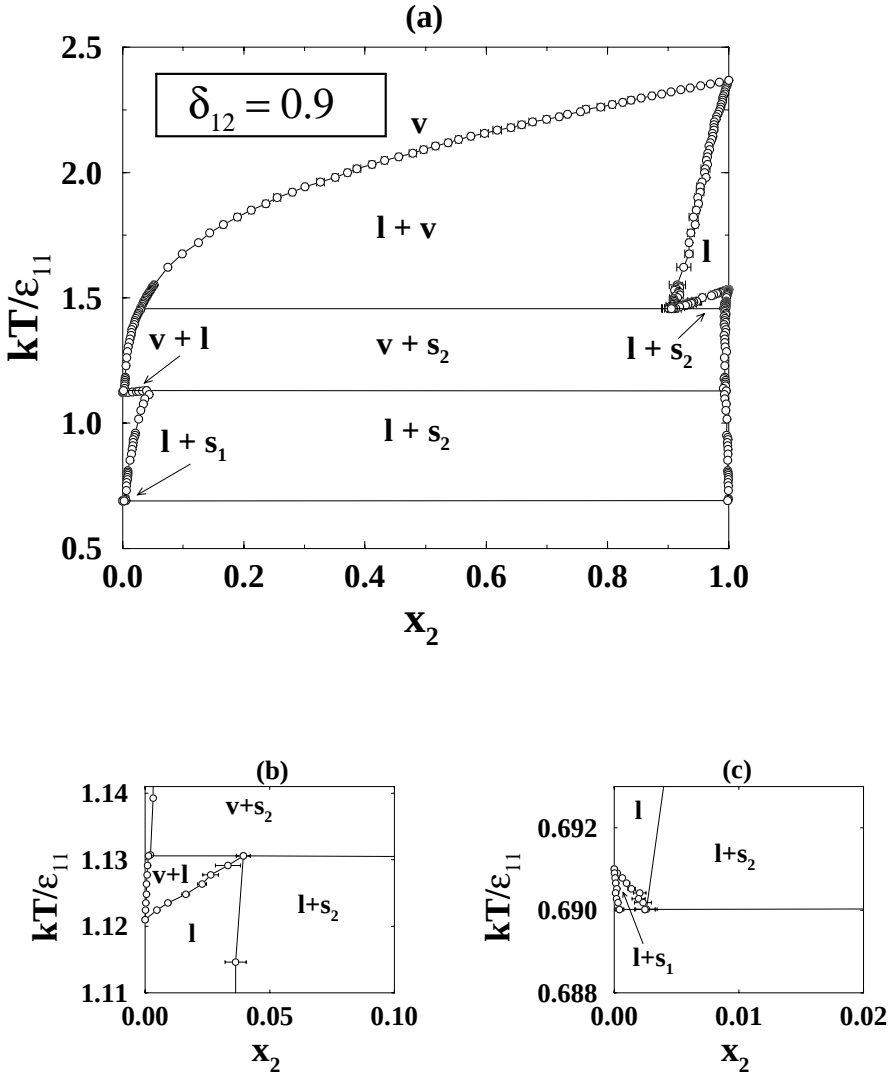

Figure 3.3: (a). Complete temperature vs. composition phase diagram at $P^* = 0.05$ for a binary Lennard-Jones mixture with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$, well-depth ratio $\epsilon_{11}/\epsilon_{22} = 0.45$, and binary interaction parameter $\delta_{12} = 0.9$. (b). Enlargement of the phase diagram near the intersection of the vapor-liquid and vapor-solid(2) regions. (c). Enlargement of the phase diagram near the intersection of the solid(1)-liquid and liquid-solid(2) regions. Symbols as in Figure 3.2.

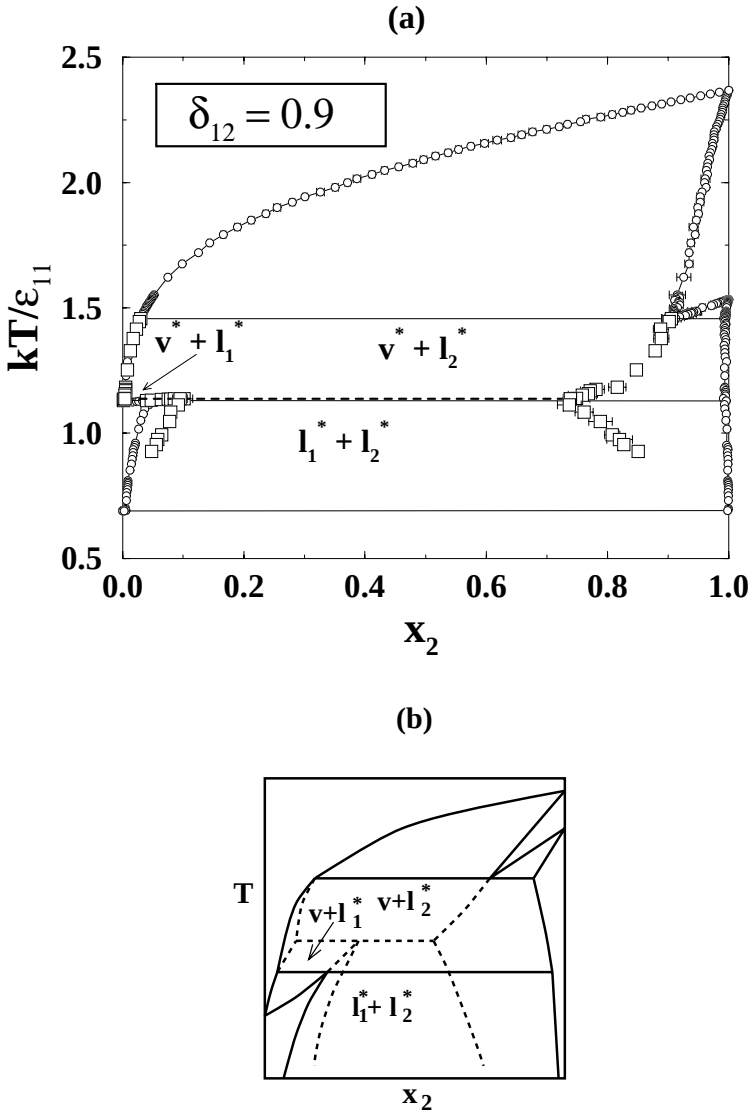

Figure 3.4: (a). Complete phase diagram shown in Figure 3.3 with the metastable vapor-liquid and liquid-liquid coexistence points (squares) determined from Gibbs-Duhem integration simulations. The dashed line indicates the metastable heteroazeotrope ($l_1 l_2 v$). Asterisks on the labels indicate that the phases identified in that region are metastable. Other symbols as in Figure 3.2. (b). Schematic phase diagram illustrating the features of the phase diagram shown in (a).

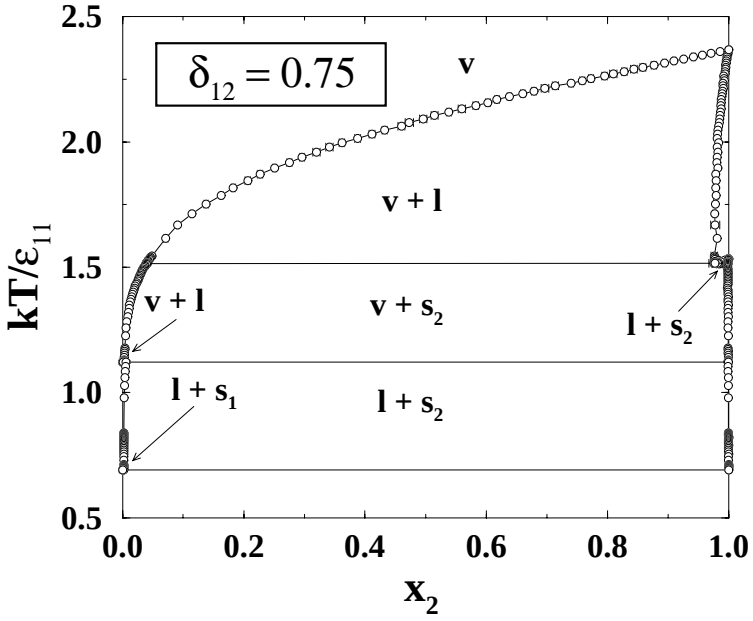

Figure 3.5: Complete temperature vs. composition phase diagram at $P^* = 0.05$ for a binary Lennard-Jones mixture with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$, well-depth ratio $\epsilon_{11}/\epsilon_{22} = 0.45$, and binary interaction parameter $\delta_{12} = 0.75$. Symbols as in Figure 3.2.

Chapter 4

The effect of pressure on complete phase behavior of binary mixtures

4.1 Introduction

Until now, we have looked at the effect of molecular size and intermolecular interactions on phase behavior for mixtures at a single pressure. In this chapter, we explore the effect of pressure on the complete phase behavior of a mixture.

The phase diagrams presented in this chapter have been calculated using the Gibbs-Duhem integration procedure described in detail in Section 2.2. The cross-species interaction parameters were calculated using the Lorentz-Berthelot[1] combining rules. In Section 4.2, we present the results of our calculation of complete $T-x$ phase diagrams for one mixture, $\sigma_{11}/\sigma_{22} = 0.85$, $\epsilon_{11}/\epsilon_{22} = 1.6$, at five reduced pressures, $P^* = 0.002$, 0.01, 0.025, 0.05, and 0.1, in order to observe how the shape of the curves on the $T-x$ diagram change with variation in pressure. In Section 4.3, we present $P-T$ projections for the mixture in Section 4.2 ($\sigma_{11}/\sigma_{22} = 0.85$, $\epsilon_{11}/\epsilon_{22} = 1.6$) and for three other mixtures with $\sigma_{11}/\sigma_{22} = 0.85$, 0.9, and 0.95, and $\epsilon_{11}/\epsilon_{22} = 0.45$,

to examine how these diagrams change with variations in Lennard-Jones diameter ratio and well-depth ratio. $P - T$ projections provide a summary of the important features of the phase behavior for a binary mixture: pure component vapor-liquid, solid-liquid, and solid-vapor coexistence curves, loci of mixture critical points, and loci of three phase coexistence points (liquid-liquid-vapor, solid-solid-liquid, etc.).

4.2 T-x phase diagrams for binary mixtures

A phase diagram for the binary Lennard-Jones mixture with $\sigma_{11}/\sigma_{22} = 0.85$ and $\epsilon_{11}/\epsilon_{22} = 1.6$ at $P^* = 0.002$ is shown in Figure 4.1. On this phase diagram vapor-liquid coexistence lines originate from pure component 1 ($x_2 = 0, T^* \equiv kT/\epsilon_{11} = 0.732$) and decrease in temperature with increasing fugacity fraction ξ_2. Solid(1)-liquid coexistence lines originate from pure component 1 ($x_2 = 0, T^* = 0.687$) and decrease in temperature with increasing fugacity fraction ξ_2. The liquid-vapor and solid(1)-liquid curves meet at $T^* = 0.6785$ and form a three-phase, solid(1)-liquid-vapor equilibrium line. Solid(1)-vapor coexistence lines originate from this three-phase line and decrease in temperature with increasing fugacity fraction ξ_2. Liquid-vapor coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 0.505$) and increase in temperature with decreasing fugacity fraction ξ_2. The liquid-vapor and solid(1)-vapor curves meet at $T^* = 0.516$ and form another three-phase, solid(1)-liquid-vapor equilibrium line. Solid(1)-liquid lines originate from this three-phase line and decrease in temperature with increasing fugacity fraction ξ_2. Liquid-solid(2) coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 0.429$) and decrease

in temperature with decreasing fugacity fraction ξ_2. The solid(1)-liquid and liquid-solid(2) curves meet at $T^* = 0.420$ and form a eutectic. Below this temperature solid(1) and solid(2) are in equilibrium (not shown).

The phase diagram for $P^* = 0.01$ is shown in Figure 4.2. Here the vapor-liquid coexistence lines have shifted to higher temperatures and they are detached from the solid-liquid coexistence region. On this phase diagram vapor-liquid coexistence lines originate from pure component 1 ($x_2 = 0, T^* = 0.884$) and decrease in temperature with increasing fugacity fraction ξ_2 until they reach the vapor-liquid coexistence temperature for pure component 2 ($x_2 = 1, T^* = 0.633$). A miscible liquid phase exists below the vapor-liquid curves. Solid(1)-liquid coexistence lines originate from pure component 1 ($x_2 = 0, T^* = 0.688$) and decrease in temperature with increasing fugacity fraction ξ_2. Liquid-solid(2) coexistence lines originate from pure component 2 ($x_2 = 1, T^* = 0.431$) and decrease in temperature with decreasing fugacity fraction ξ_2. The solid(1)-liquid and liquid-solid(2) curves meet at $T^* = 0.422$ and form a eutectic.

The phase diagram for $P^* = 0.025$ is shown in Figure 4.3. This phase diagram is similar to the one for $P^* = 0.01$ shown in Figure 4.2, except that the vapor-liquid coexistence region appears at higher temperatures. The solid(1)-liquid and liquid-solid(2) curves meet at $T^* = 0.424$ and form a eutectic.

The phase diagram for $P^* = 0.05$ is shown in Figure 4.4. The vapor-liquid coexistence region has again shifted to higher temperatures but because this pressure is above the critical pressure for pure component 2 ($P_c^* = 0.0476$) the coexistence

region ends at a mixture critical point instead of spanning the diagram from x_2 = 0.0 to x_2 = 1.0. We estimate to be T^* = 0.91 and x_2 = 0.7. The solid(1)-liquid and liquid-solid(2) curves meet at T^* = 0.425 and form a eutectic.

The phase diagram for P^* = 0.1 is shown in Figure 4.5. Attempts to calculate a vapor-liquid coexistence curve originating from pure component 1 ($x_2 = 0, T^* =$ 1.261) were unsuccessful because the liquid phase vaporized as we stepped away from x_2 = 0 (ξ_2 = 0) during the Gibbs-Duhem integration. This is not surprising since this coexistence point is very near the critical temperature and pressure of pure component 1 (T_c^* = 1.316, P_c^* = 0.124). The solid(1)-liquid and liquid-solid(2) curves meet at T^* = 0.432 and form a eutectic.

To understand how variations in pressure lead to the complete phase diagrams displayed in Figures 4.1- 4.5, it is helpful to consider the schematic phase diagram shown in Figure 4.6. The pure component solid-liquid coexistence temperatures and the eutectic temperature and composition are essentially unaffected by changes in pressure, while the pure component vapor-liquid coexistence temperatures increase with increasing pressure. This is not surprising since liquids and solids are relatively incompressible and hence, their properties are not strongly affected by pressure. At P^* = 0.002, there is interference between the vapor-liquid and solid-liquid coexistence regions. As the pressure increases to P^* = 0.01, the pure component vapor-liquid coexistence temperatures increase, shifting the entire vapor-liquid coexistence region to higher temperatures until it lifts off the solid-liquid coexistence region. At P^* = 0.05, we begin to enter the critical region for mixtures rich in com-

ponent 2 and the vapor-liquid coexistence region begins to shrink away from the x_2 = 1 axis. Finally, at P^* = 0.1, the entire range of mixture compositions are in the critical region and the vapor-liquid coexistence region has disappeared.

4.3 P-T projections for binary mixtures

In this section we present $P-T$ projections for Lennard-Jones binary mixtures. To construct a $P-T$ projection, we first plot the pure component coexistence data (taken from the work of Kofke and coworkers[2,3]): vapor-liquid, solid-liquid, and solid-vapor equilibrium curves, vapor-liquid critical points, and solid-liquid-vapor triple points. We then use Gibbs-Duhem integration to calculate $T-x$ diagrams at several pressures and plot all of the three-phase coexistence (s_1s_2l, s_2lg, s_1lg) temperatures and pressures on the $P-T$ diagram. To illustrate this, the results for the binary mixture with σ_{11}/σ_{22} = 0.85 and $\epsilon_{11}/\epsilon_{22}$ = 1.6 described in Section 4.3 are given on the $P-T$ projection shown in Figure 4.7. The eutectic (s_1s_2l) and solid(1)-liquid-vapor (s_1lg) temperatures determined from the simulations in this work are shown by open circles, with dot-dashed lines connecting the points to guide the eye. The eutectic (s_1s_2l) locus is relatively independent of pressure. The solid(1)-liquid-vapor (s_1lg) locus originates from the triple point of pure component 1 and passes through a maximum pressure at P^* = 0.005. We anticipate that the s_1lg locus will terminate at the quadruple point (s_1s_2lg) of the mixture[4], which is below P^* = 0.002 (the lowest pressure considered here). Our estimate of the binary mixture critical point at P^* = 0.05 is shown by an asterisk.

In the same manner, we constructed $P-T$ phase diagrams for binary mixtures with $\epsilon_{11}/\epsilon_{22} = 0.45$ and $\sigma_{11}/\sigma_{22} = 0.85$, 0.90, and 0.95, using results from $T-x$ calculations at $P^* = 0.002$, 0.025, and 0.05. The $P-T$ diagrams for these mixtures are shown in Figures 4.8 - 4.10. In Figure 4.8, for the binary mixture with $\sigma_{11}/\sigma_{22} = 0.85$ and $\epsilon_{11}/\epsilon_{22} = 0.45$, the eutectic ($s_1 s_2 l$) locus is relatively independent of pressure and can be found at approximately $T^* = 0.655$. Two branches of the $s_2 lg$ locus are shown. The first branch originates from the triple point of pure component 2 and decreases in temperature with increasing pressure. The second branch starts from $P^* = 0.002$ and increases in temperature with increasing pressure to $P^* = 0.025$. Again, we expect that this locus will form a smooth curve that originates from the triple point of the less volatile component (species 2) and passes through a maximum pressure before terminating at the quadruple point ($s_1 s_2 lg$) of the mixture[4]. Based on the fact that there was no interference between vapor-liquid and solid-liquid equilibria at $P^* = 0.05$, we can conclude that between $P^* = 0.025$ and 0.05, these two branches will meet at a pressure maximum corresponding to the pressure at which the vapor-liquid and solid-liquid coexistence regions last interfere with each other. A similar pattern of phase behavior was observed for the mixtures with $\sigma_{11}/\sigma_{22} = 0.90$ and 0.95, shown in Figures 4.9 and 4.10. Thus, we see that as the diameter ratio increases, the vapor-liquid and solid-liquid interference continues to higher pressures and the temperature difference between the two branches increases. Additionally, the locus of solid-solid-liquid temperatures shifts from temperatures below both pure component melting points at $\sigma_{11}/\sigma_{22} = 0.85$ to temperatures above the pure

component 1 melting point at $\sigma_{11}/\sigma_{22} = 0.95$, changing the characteristic shape of the solid-liquid coexistence region from eutectic to peritectic.

Upon comparing the phase diagrams shown in Figures 4.7 and 4.8 for the mixtures with $\sigma_{11}/\sigma_{22} = 0.85$ and $\epsilon_{11}/\epsilon_{22} = 0.45$ and 1.6, we find that decreasing the well-depth ratio causes the coexistence curves for pure component 2 to shift from temperatures lower than those for pure component 1 to temperatures higher than those for pure component 1. At $\epsilon_{11}/\epsilon_{22} = 1.6$, the vapor-liquid and solid-liquid interference ends between $P^* = 0.002$ and 0.01, but for $\epsilon_{11}/\epsilon_{22} = 0.45$, the vapor-liquid and solid-liquid interference ends between $P^* = 0.025$ and 0.05. Thus, we conclude that as $\epsilon_{11}/\epsilon_{22}$ decreases the maximum pressure in the slg locus decreases.

4.4 Summary

The Gibbs-Duhem integration technique was combined with semigrand canonical Monte Carlo simulations to calculate complete phase diagrams for binary Lennard-Jones mixtures at several different pressures to explore the effects of molecular size, intermolecular attractions, and pressure on the complete phase behavior of a mixture. We calculated $T - x$ phase diagrams for binary Lennard-Jones mixtures with diameter ratios ranging from 0.85 – 0.95 and attractive well-depth ratios of 0.45 and 1.6 at reduced pressures $P^* = 0.002$, 0.01, 0.025, 0.05, and 0.1. We then constructed a $P - T$ phase diagram for each mixture.

4.5 References

[1] J. S. Rowlinson and F. L. Swinton, *Liquids and Liquid Mixtures*. (Butterworth Scientific, London, 1982).

[2] D. A. Kofke, Molecular Physics, **78**(6), 1331–1336 (1993).

[3] R. Agrawal and D. A. Kofke, Molecular Physics, **85**(1), 43–59 (1995).

[4] edited by A. FindlayA. N. Campbell and N. O. Smith, *The phase rule and its applications*. (Dover Publications, New York, ninth edition, 1951).

4.6 Figures

Page

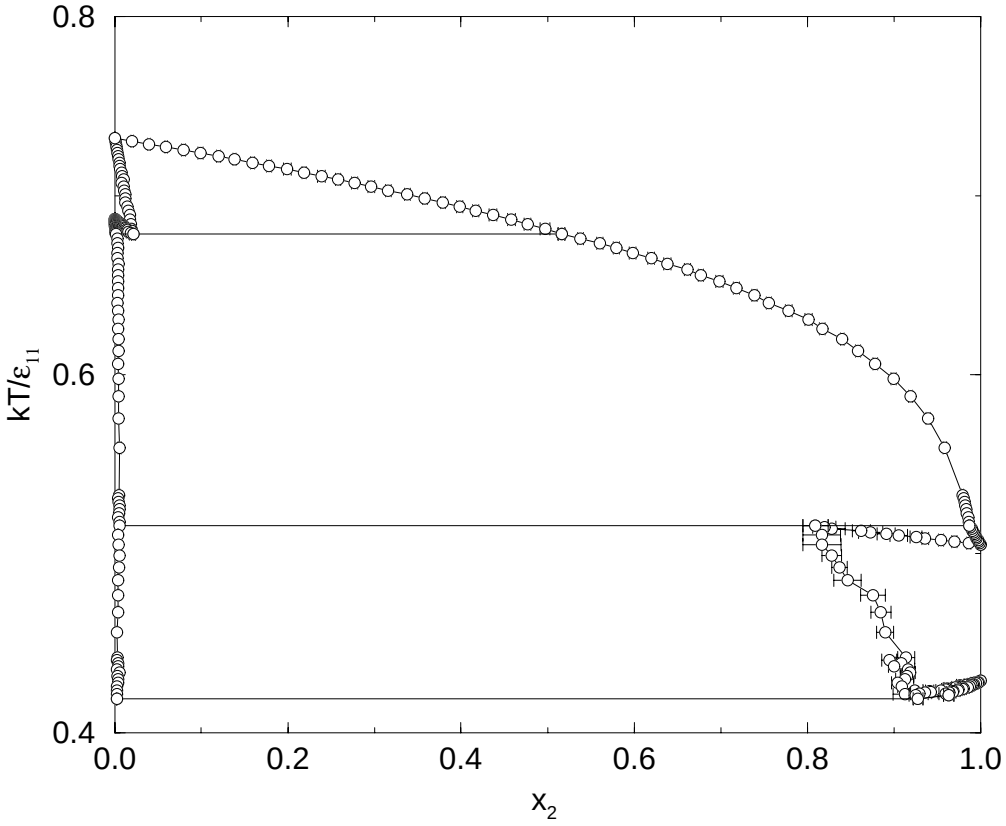

Figure 4.1: Temperature vs. composition phase diagram for a Lennard-Jones binary mixture with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$ and $\epsilon_{11}/\epsilon_{22} = 1.6$ at $P^* = 0.002$. The circles represent data from Gibbs-Duhem integration simulations. Error bars are shown when they are larger than the width of the symbol. Lines are drawn through the points for clarity.

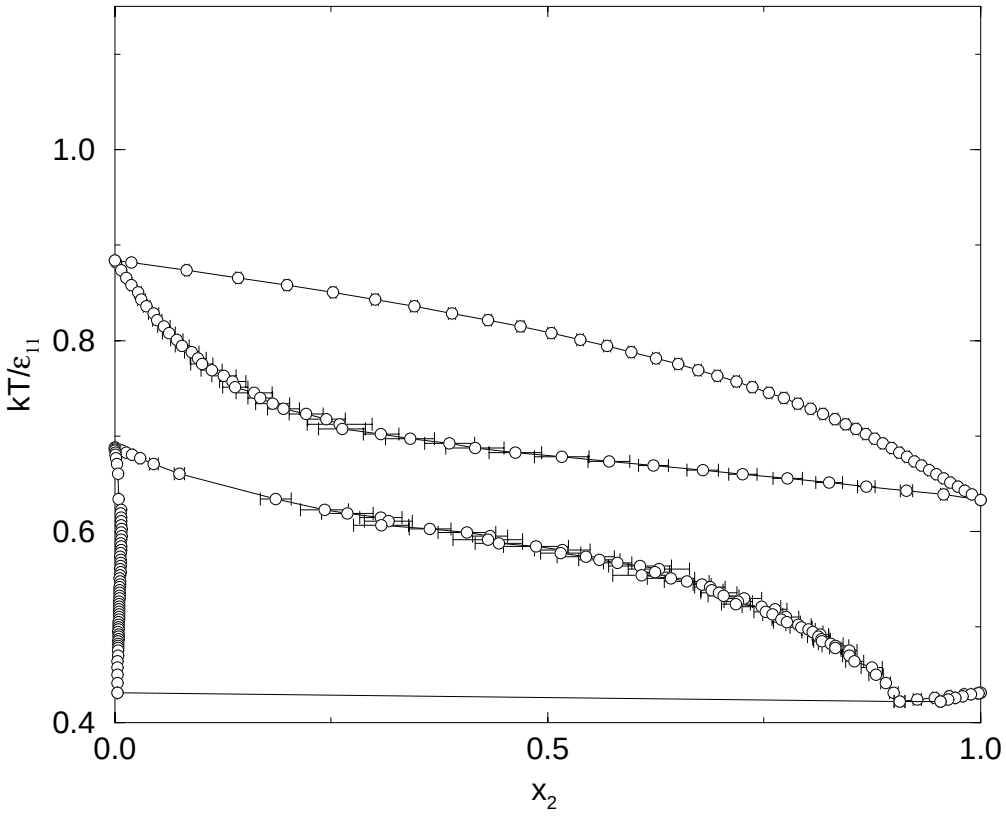

Figure 4.2: Temperature vs. composition phase diagram for a Lennard-Jones binary mixture with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$ and $\epsilon_{11}/\epsilon_{22} = 1.6$ at $P^* = 0.01$. Symbols as in Figure 4.1

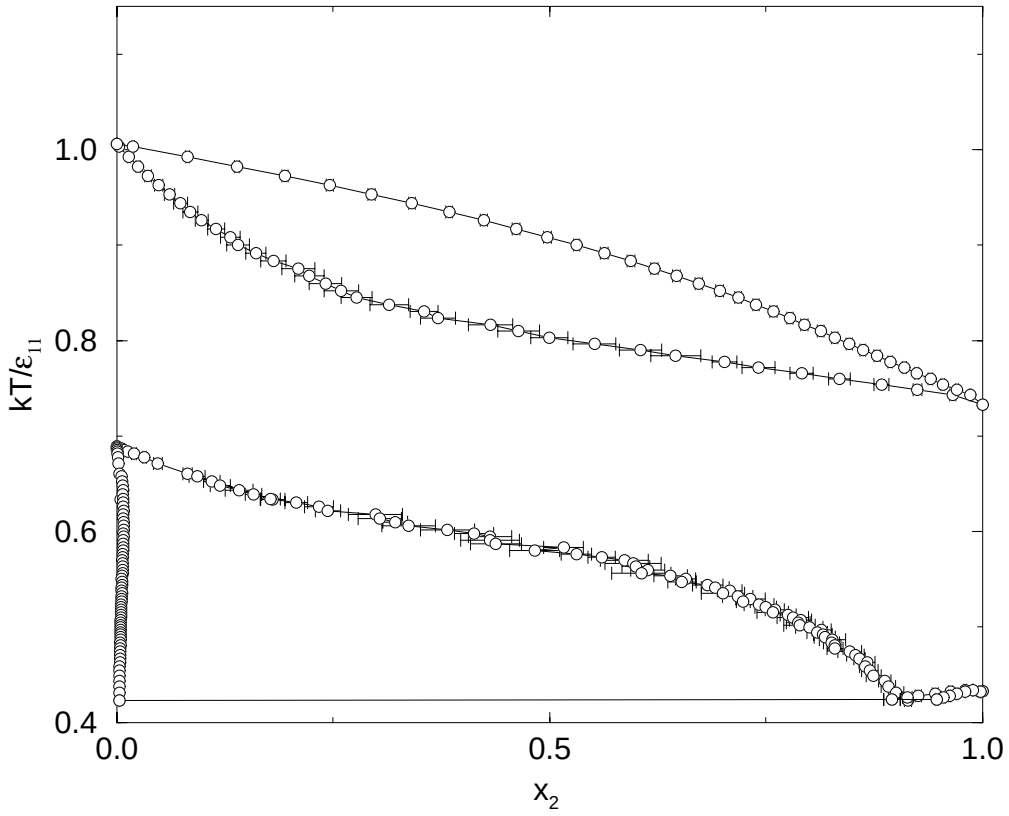

Figure 4.3: Temperature vs. composition phase diagram for a Lennard-Jones binary mixture with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$ and $\epsilon_{11}/\epsilon_{22} = 1.6$ at $P^* = 0.025$. Symbols as in Figure 4.1

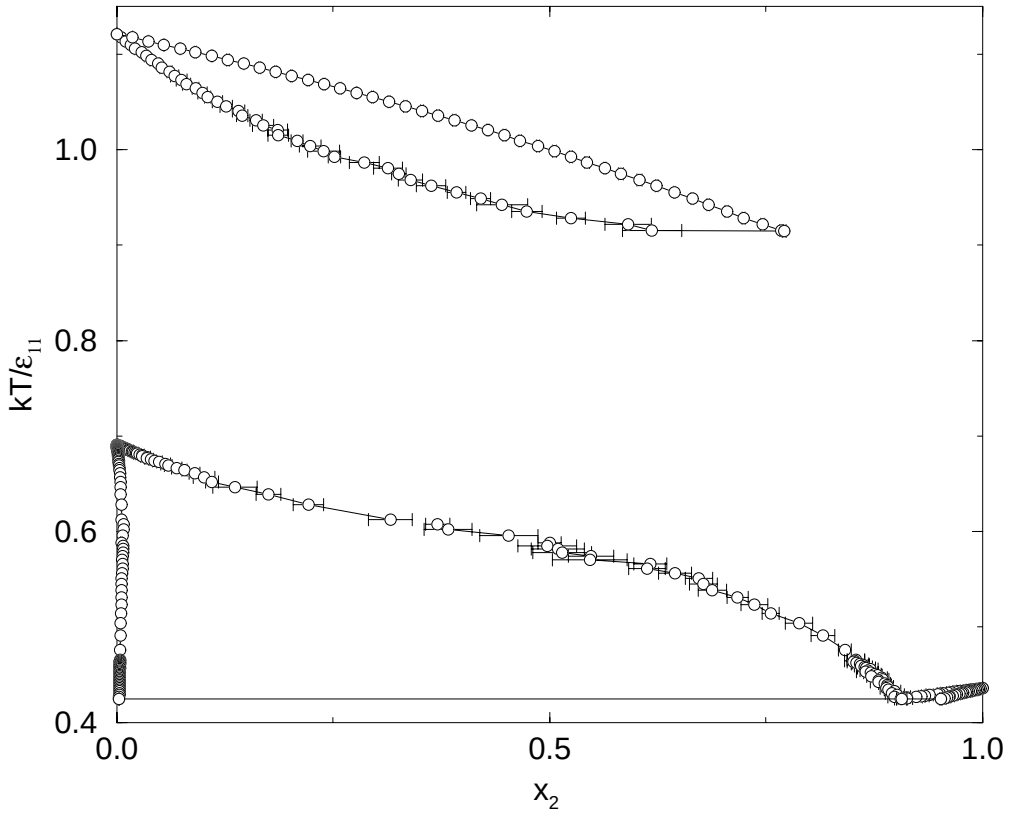

Figure 4.4: Temperature vs. composition phase diagram for a Lennard-Jones binary mixture with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$ and $\epsilon_{11}/\epsilon_{22} = 1.6$ at $P^* = 0.05$. Symbols as in Figure 4.1

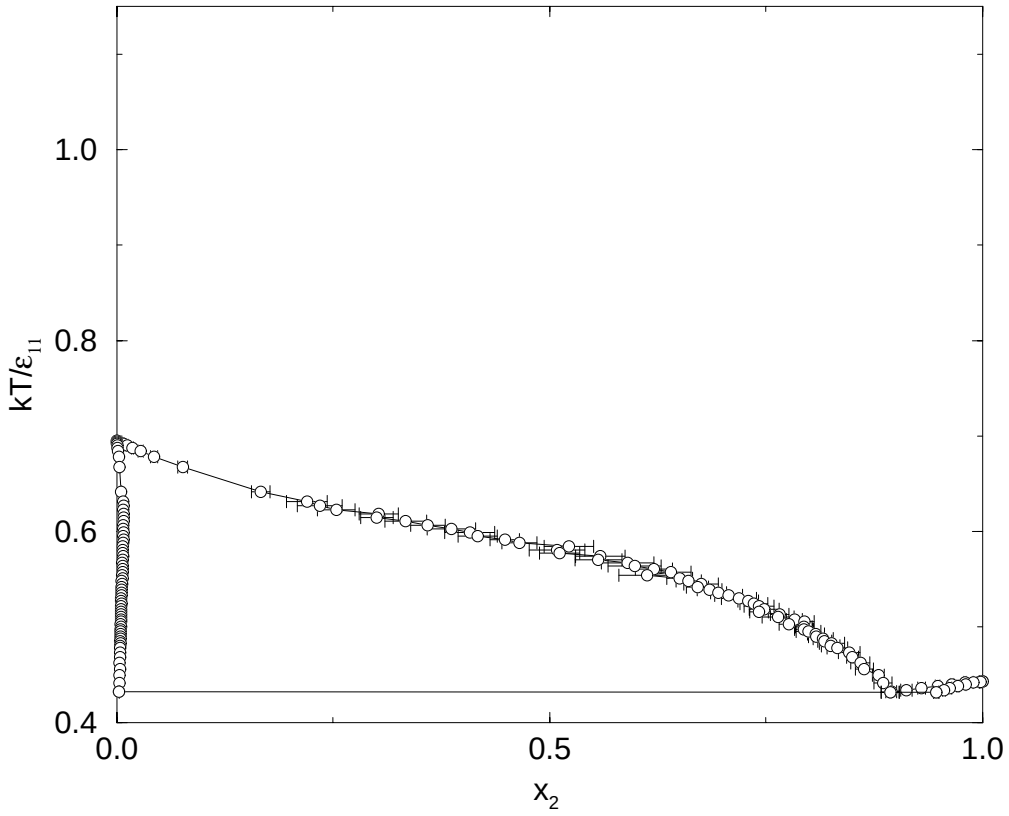

Figure 4.5: Temperature vs. composition phase diagram for a Lennard-Jones binary mixture with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$ and $\epsilon_{11}/\epsilon_{22} = 1.6$ at $P^* = 0.1$. Symbols as in Figure 4.1

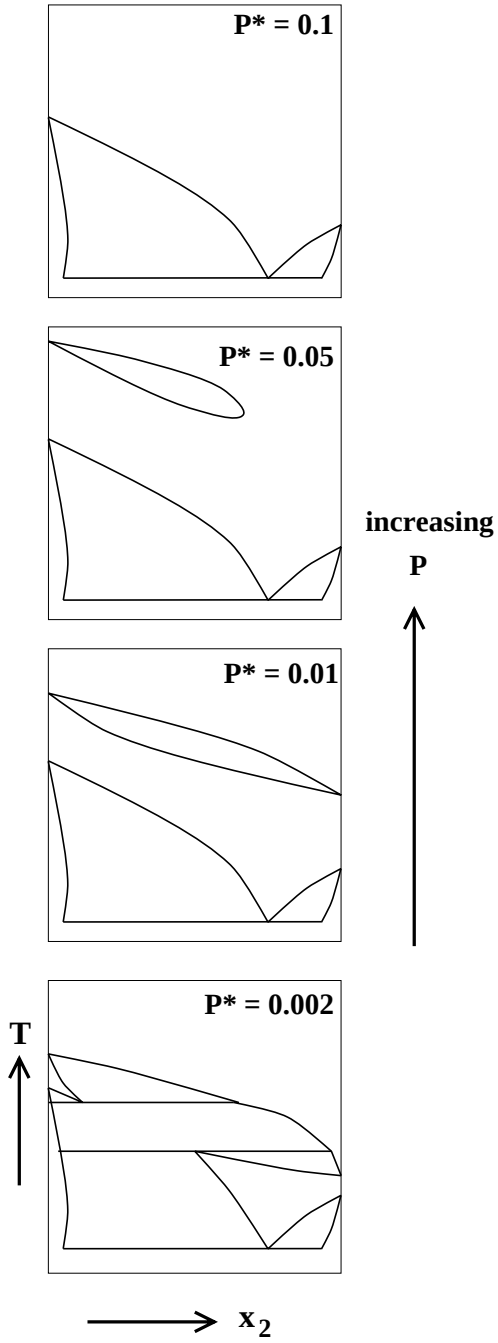

Figure 4.6: Schematic temperature versus composition phase diagrams illustrating the effect of increasing pressure for the binary Lennard-Jones mixture with σ_{11}/σ_{22} = 0.85 and $\epsilon_{11}/\epsilon_{22}$ = 1.6.

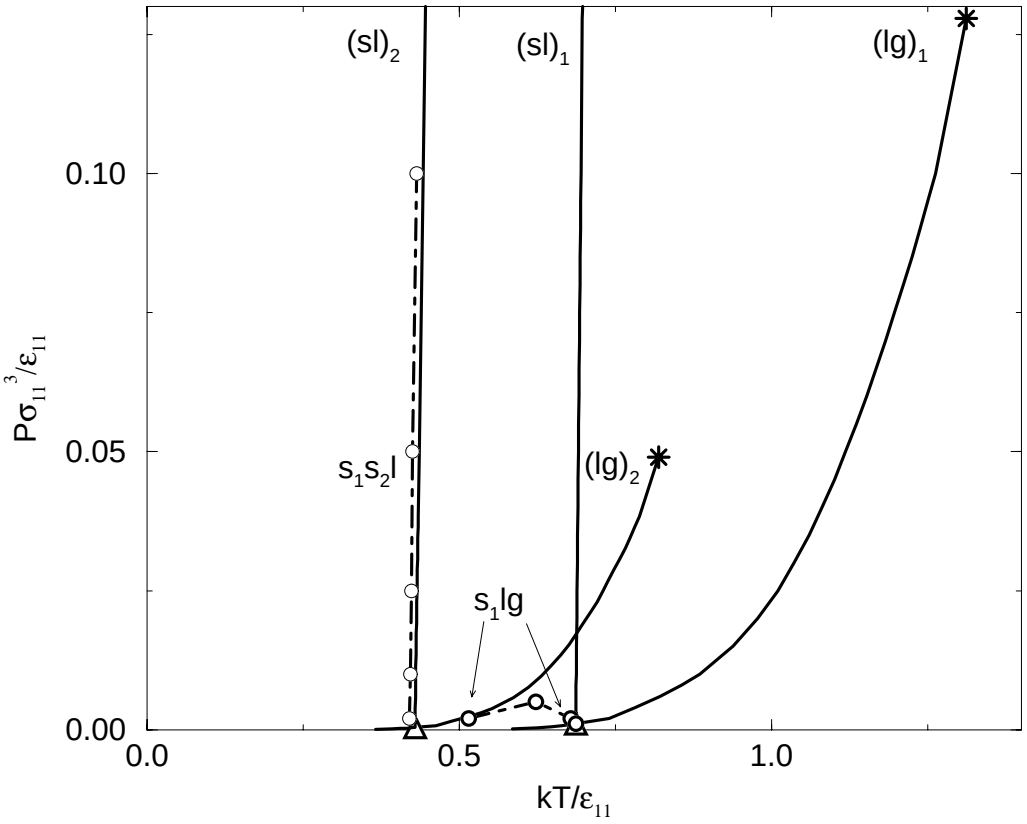

Figure 4.7: Pressure-temperature projection for a Lennard-Jones binary mixture with diameter ratio $\sigma_{11}/\sigma_{22} = 0.85$ and well-depth ratio $\epsilon_{11}/\epsilon_{22} = 1.6$. The open circles represent three-phase coexistence data from Gibbs-Duhem integration simulations. Dot-dashed lines are drawn through the simulation points as a guide to the eye. The remaining symbols are as follows: solid lines represent the pure component vapor pressure and melting curves, asterisks denote pure component and mixture vapor-liquid critical points, open triangles denote pure component triple points.

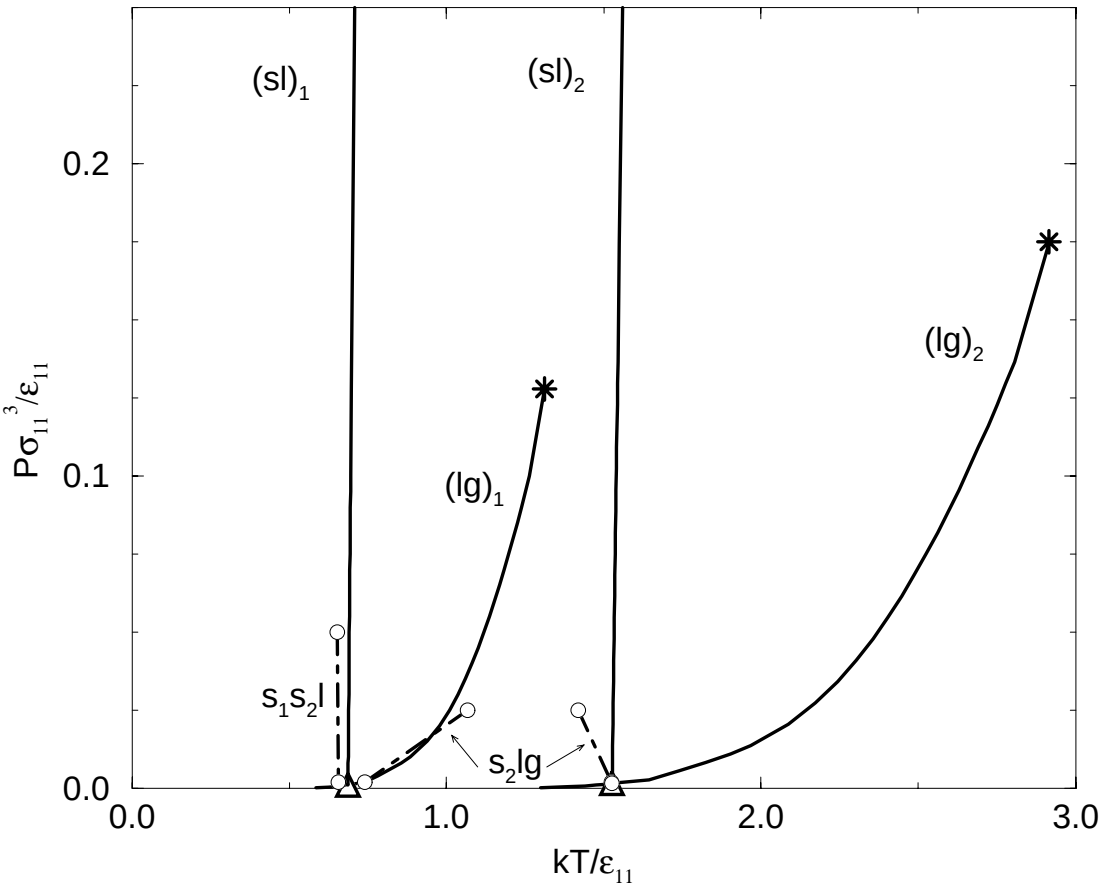

Figure 4.8: Pressure-temperature projection for a Lennard-Jones binary mixture with diameter ratio σ_{11}/σ_{22} = 0.85 and well-depth ratio $\epsilon_{11}/\epsilon_{22}$ = 0.45. Symbols as in Figure 4.7.

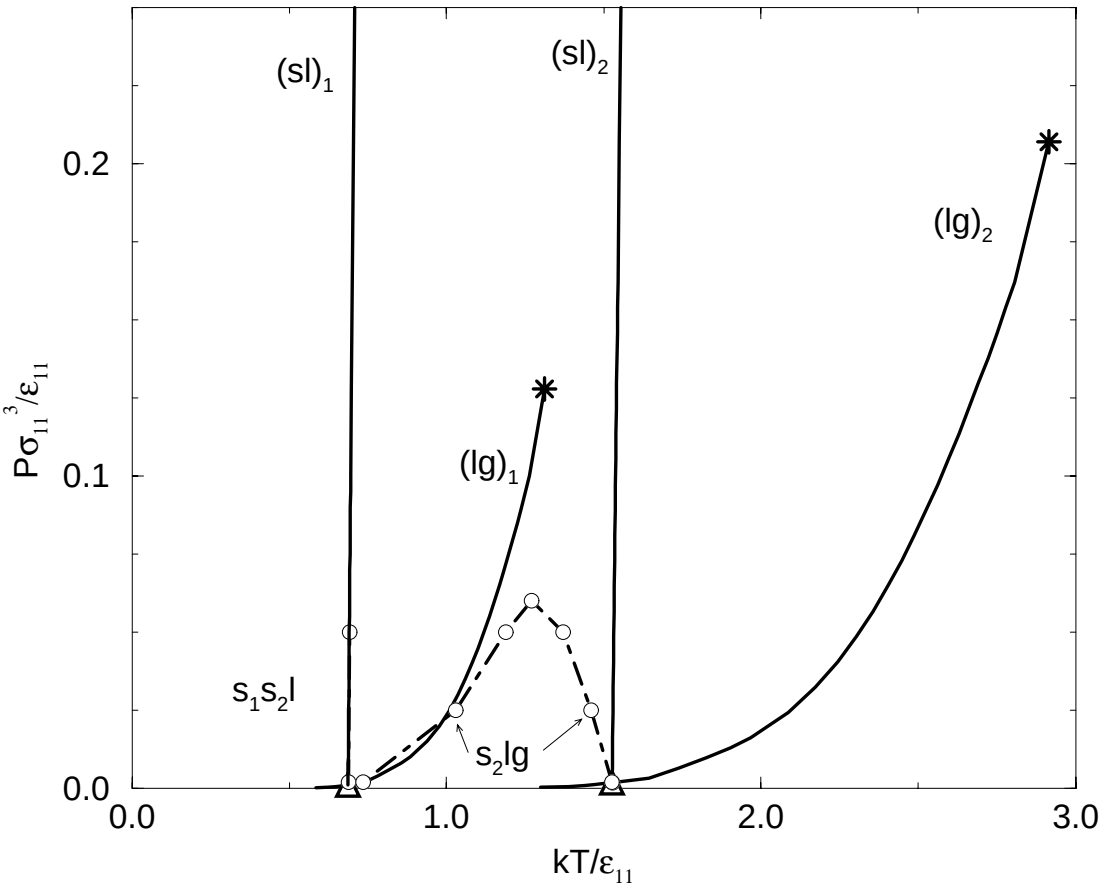

Figure 4.9: Pressure-temperature projection for a Lennard-Jones binary mixture with diameter ratio σ_{11}/σ_{22} = 0.90 and well-depth ratio $\epsilon_{11}/\epsilon_{22}$ = 0.45. Symbols as in Figure 4.7.

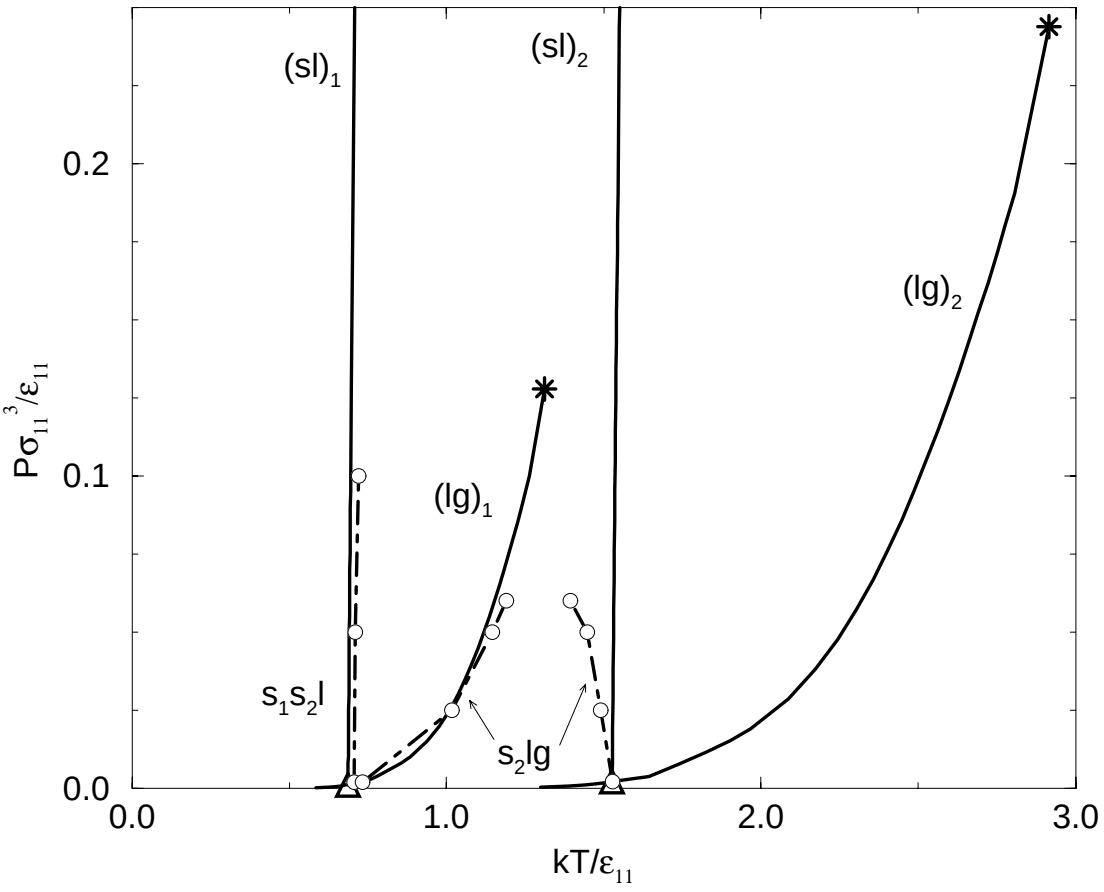

Figure 4.10: Pressure-temperature projection for a Lennard-Jones binary mixture with diameter ratio $\sigma_{11}/\sigma_{22} = 0.95$ and well-depth ratio $\epsilon_{11}/\epsilon_{22} = 0.45$. Symbols as in Figure 4.7.

CHAPTER 5

PRESSURE-TEMPERATURE PROJECTION FOR A BINARY LENNARD-JONES MIXTURE

5.1 Introduction

Phase equilibria for binary mixtures of spherically symmetric molecules has been the subject of intensive investigation for decades. Van Konynenburg and Scott[1,2] analyzed and classified the phase diagrams predicted by the van der Waals equation of state for binary mixtures. Remarkably, the simple van der Waals equation of state was found to exhibit five of the six types of fluid phase behavior observed experimentally. This landmark study has been followed by similar analyses for other equations of state, such as the Redlich-Kwong[3] and Carnahan-Starling-Redlich-Kwong[4] equations of state.

Most of the research directed at understanding how intermolecular interactions affect phase behavior has focused exclusively on fluid phase equilibria. However, in real systems solid phases form and often interrupt the complex fluid phase behavior[5]. In order to obtain a better overall picture of mixture phase behavior, all types

of equilibrium between gas, liquid and solid phases must be considered. Valyashko[6] classified complete phase diagrams for binary mixtures based on experimental results. Garcia and Luks[7] recently examined complete phase diagrams for binary mixtures using the van der Waals equation of state and a simple solid state fugacity model. Molecular simulation has become a popular way to study the phase behavior of model systems[8]. Numerous simulations investigating binary mixture phase behavior for the Lennard-Jones intermolecular potential, the quintessential model of a spherically symmetric molecule, have been conducted for both fluid-fluid[9–13] and solid-liquid[14] phase equilibria. Vlot and coworkers[15] calculated complete phase diagrams for symmetric (equal diameters, $\sigma_{11} = \sigma_{22}$; equal attractions, $\epsilon_{11} = \epsilon_{22}$) Lennard-Jones mixtures by using Monte Carlo simulation for selected state points to determine the excess free energy as a function of composition. This free energy versus composition data was fit with a two-parameter Redlich Kister polynomial and the convex envelope construction method was used to determine the phase diagram.

In this chapter we calculate complete phase diagrams for a binary Lennard-Jones mixture at three pressures via Gibbs-Duhem integration and semigrand canonical Monte Carlo simulation. This is the first time that a complete phase diagram has been determined directly from molecular simulation. The remainder of the chapter is organized as follows. We first outline the Gibbs-Duhem integration method and describe how we applied the technique to the calculation of a complete phase diagram. We then present a complete phase diagram for a binary Lennard-Jones mixture.

5.2 Method

The coexistence lines were calculated using Gibbs-Duhem integration[16,17]. In this method, phase coexistence is determined by integrating the Clapeyron differential equation. The Clapyeron equation for equilibrium between two binary phases (α and γ) at constant pressure is

$$\frac{d\beta}{d\xi_2} = \frac{(x_2^\alpha - x_2^\gamma)}{\xi_2(1-\xi_2)(h^\alpha - h^\gamma)}, \tag{5.1}$$

where β is the reciprocal temperature, $1/kT$, with k the Boltzmann constant and T the absolute temperature, ξ_2 is the fugacity fraction of species 2, $\xi_2 \equiv \hat{f}_2 / \sum \hat{f}_i$, with $\hat{f}_i$, the fugacity of species i in solution, x_2 is the mole fraction of species 2, and h is the molar enthalpy. The right-hand side of Eq. (5.1) can be integrated numerically to find an equation for β as a function of ξ_2 if we have an initial condition describing the temperature, fugacity fraction, enthalpies and compositions at one coexistence point.

In our work, a convenient choice for the initial coexistence condition is the vapor-liquid or solid-liquid equilibrium condition for either of the pure components. The integrand in Eq. (5.1) is undefined for pure components ($\xi_2 = 0, x_2 = 0$ and $\xi_2 = 1, x_2 = 1$) but it can be estimated using the limiting case of infinite dilution. This procedure is described in Section 2.2 and will not be repeated here. The enthalpies and mole fractions needed for the integration of Eq. (5.1) were obtained by semigrand canonical Monte Carlo simulation (constant temperature, pressure, total number of

molecules, and fugacity fraction) of the two phases. In this work, the simulations were run with a system size of 500 particles per phase. The pressure was held constant. The temperature and fugacity fraction were varied according to the values specified by the predictor-corrector algorithm used to numerically integrate Eq. (5.1).

In this mixture, we encountered occurrences of three-phase lines, such as is found for a heteroazeotrope (liquid-liquid-vapor). In this case, two Gibbs-Duhem integrations were conducted, each starting from the vapor-liquid coexistence condition of the pure components ($\xi_2 = 0$ and $\xi_2 = 1$). At some ξ_2 (unknown at the commencement of the two integrations) the vapor phase coexistence lines will cross, thus, determining the temperature, fugacity fraction, and coexistence compositions of the three coexisting phases: liquid(1), liquid(2), and vapor. The liquid phase mole fractions and enthalpies then become the initial condition for the liquid-liquid coexistence curve found below the heteroazeotrope temperature.

5.3 Results

Using Gibbs-Duhem integration and semigrand canonical Monte Carlo simulation we calculated complete phase diagrams for a binary Lennard-Jones mixture with the following parameters: $\sigma_{11} = \sigma_{22} = 1.0$, $\sigma_{12} = 0.85$, $\epsilon_{11} = 1.0$, $\epsilon_{12} = 0.82$, and $\epsilon_{22} = 1.2$, at reduced pressures $P^* \equiv P\sigma_{11}^3/\epsilon_{11} = 0.007$, 0.05, and 0.1. In Fig. 1, we present the complete $T - x$ phase diagram for $P^* = 0.05$. On this phase diagram, vapor-liquid coexistence curves originate from both $x_2 = 0$ ($\xi_2 = 0$) and $x_2 = 1$ ($\xi_2 = 1$). Both sets of curves decrease in temperature as fugacity fraction is varied

until they meet and form a heteroazeotrope at reduced temperature, $T^* \equiv kT/\epsilon_{11} = 1.07$. At this temperature three phases coexist: a liquid mixture rich in component 1 (liquid(1)), a vapor mixture, and a liquid mixture rich in component 2 (liquid(2)). Below this temperature liquid(1) and liquid(2) are in equilibrium.

At $T^* = 0.828$, the solid and liquid phases of pure component 2 are in equilibrium. Solid-liquid coexistence curves originate from $x_2 = 1$ ($\xi_2 = 1$) and decrease in temperature as fugacity fraction is varied until they meet the liquid(1)-liquid(2) coexistence curves and form a monotectic at $T^* = 0.802$. At this temperature, three phases coexist: liquid(1), liquid(2), and a solid mixture rich in component 2 (solid(2)). Below this temperature, liquid(1) and solid(2) are in equilibrium.

At $T^* = 0.691$, the solid and liquid phases of pure component 1 are in equilibrium. Solid-liquid coexistence curves originate from $x_2 = 0$ ($\xi_2 = 0$) and decrease in temperature as fugacity fraction is varied until they meet the liquid(1)-solid(2) coexistence curves to form a eutectic at $T^* = 0.689$. At this temperature, three phases coexist: solid mixture rich in component 1 (solid(1)), liquid(1), and solid(2). Below this temperature, solid(1) and solid(2) are in equilibrium.

Similar $T - x$ phase diagrams were obtained for $P^* = 0.007$ and 0.1, i.e., the phase diagrams had heteroazeotrope, monotectic, and eutectic lines. The results obtained from the Gibbs-Duhem integration calculations at the three pressures are given on the P-T projection shown in Fig. 2. The heteroazeotrope ($l_1 l_2 g$), monotectic ($l_1 l_2 s$), and eutectic ($s_1 s_2 l$) temperatures determined in this work are shown by open circles. The pure component vapor pressure (lg) and melting curves (sl), critical

points (filled circles), and triple points (open triangles) are based on the simulation data of Kofke and coworkers[18,17].

Fluid phase coexistence points for this mixture have also been calculated by Canongia Lopes[9] using Gibbs ensemble Monte Carlo. They found a locus of heteroazeotrope ($l_1 l_2 g$) temperatures that coincides with our heteroazeotrope locus but disappears above T^* = 1.1 and below T^*= 0.9. Canongia Lopes interpreted the disappearance of liquid(1)-liquid(2) immiscibility above T^* = 1.1 and below T^* = 0.9 as evidence for the presence of upper and lower critical endpoints, respectively. Based on these inferred upper and lower critical endpoints, Canongia Lopes concluded that this mixture displays Type VI behavior (closed-loop liquid-liquid immiscibility). Canongia Lopes does not present enough state points for us to be able to determine why our results differ. Thus, we cannot, at the present time, offer a satisfactory explanation for the discrepancy.

5.4 Summary

The Gibbs-Duhem integration technique was combined with semigrand canonical Monte Carlo simulations to calculate complete T-x phase diagrams for a binary Lennard-Jones mixture. Complete phase diagram calculations for other classes of binary mixtures are currently underway.

5.5 References

[1] R. L. Scott and P. H. van Konynenburg, Discussions of the Faraday Society, **49**, 87–97 (1970).

[2] P. H. van Konynenburg and R. L. Scott, Philosophical Transactions of the Royal Society of London A., **298**, 495–540 (1980).

[3] U. K. Deiters and I. L. Pegg, Journal of Chemical Physics, **90**, 6632–6641 (1989).

[4] T. Kraska and U. K. Deiters, Journal of Chemical Physics, **96**, 539–547 (1992).

[5] G. M. Schneider, in *Chemical Thermodynamics* edited by M. L. McGlashan (The Chemical Society, London, 1978).

[6] V. M. Valyashko, Z. Phys. Chemie, Leipzig, **267**(3), 481–493 (1986).

[7] D. C. Garcia and K. D. Luks, Fluid Phase Equilibria, **161**, 91–106 (1999).

[8] A. Z. Panagiotopoulos, Journal of Physics: Condensed Matter, **12**, R25–R52 (2000).

[9] J. N. C. Lopes, Molecular Physics, **96**(11), 1649–1658 (1999).

[10] A. M. Georgolaki, I. V. Ntouros, D. P. Tassios, and A. Z. Panagiotopoulos, Fluid Phase Equilibria, **100**, 153–170 (1994).

[11] M. Guo, Y. Li, Z. Li, and J. Lu, Fluid Phase Equilibria, **98**, 129–139 (1994).

[12] V. I. Harismiadis, N. K. Kourtras, D. P. Tassios, and A. Z. Panagiotopoulos, Fluid Phase Equilibria, **65**, 1–18 (1991).

[13] M. E. van Leeuwen, C. J. Peters, J. de Swaan Arons, and A. Z. Panagiotopoulos, Fluid Phase Equilibria, **66**, 57–75 (1991).

[14] M. R. Hitchcock and C. K. Hall, Journal of Chemical Physics, **110**(23), 11433–11444 (1999).

[15] M. J. Vlot, J. C. van Miltenburg, H. A. J. Oonk, and J. P. van der Eerden, Journal of Chemical Physics, **107**(23), 10102–10111 (1997).

[16] D. A. Kofke, Molecular Physics, **78**(6), 1331–1336 (1993).

[17] D. A. Kofke, Journal of Chemical Physics, **98**(5), 4149–4162 (1993).

[18] R. Agrawal and D. A. Kofke, Molecular Physics, **85**(1), 43–59 (1995).

5.6 Figures

Page

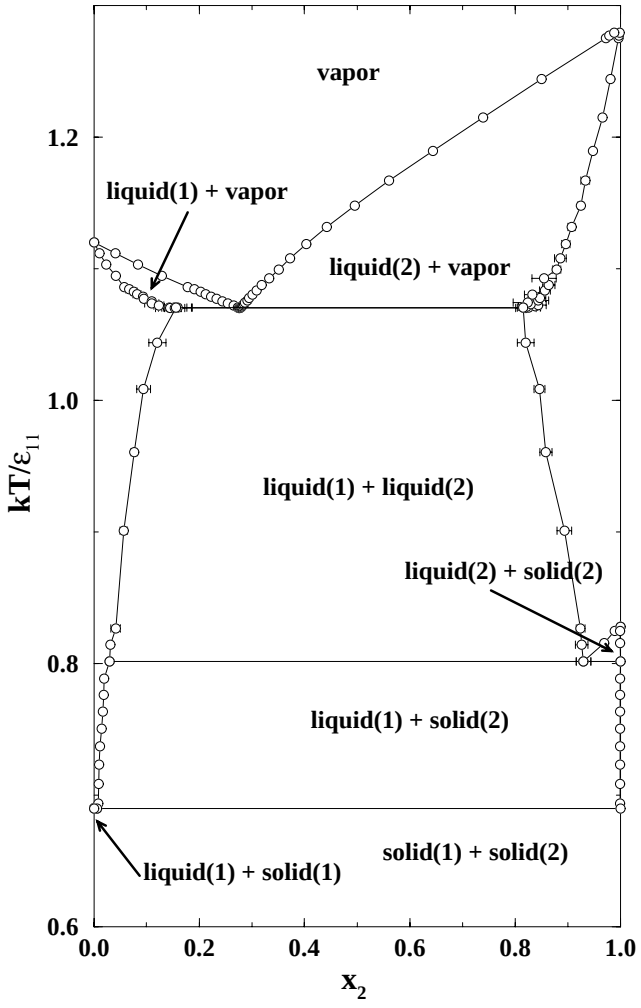

Figure 5.1: Complete $T - x$ phase diagram at P^*= 0.05. The Lennard-Jones parameters are given in the text. The open circles represent data from Gibbs-Duhem integration simulations. Error bars are shown when they are larger than the width of the symbol. Lines are drawn through the simulation points for clarity.

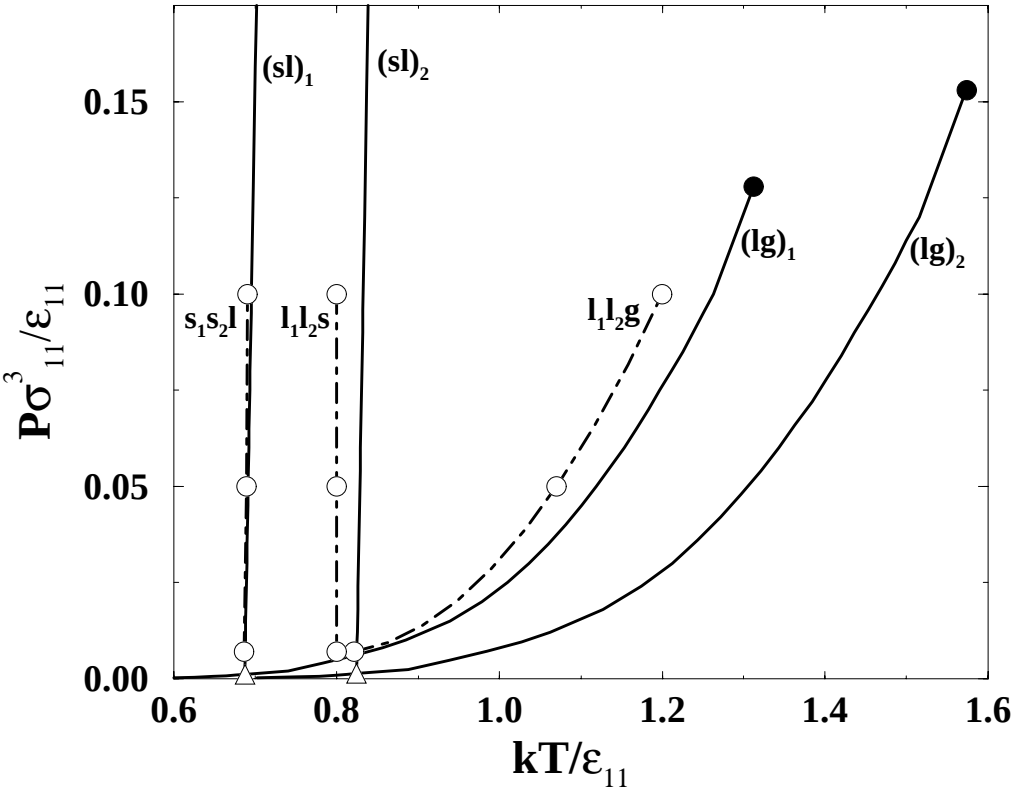

Figure 5.2: $P - T$ projection summarizing the solid (s), liquid (l), and vapor (g) phase coexistence lines for the binary Lennard-Jones mixture considered in this work. The open circles represent three-phase coexistence data from Gibbs-Duhem integration simulations. Dot-dashed lines are drawn through the simulation points for clarity. The remaining symbols are as follows: solid lines represent the pure component vapor pressure and melting curves, filled circles denote pure component vapor-liquid critical points, open triangles denote pure component triple points.

Chapter 6

Future Work

In this chapter we first comment on the benefits of using the Gibbs-Duhem integration method to calculate complete phase diagrams and then make some recommendations for future work.

The calculations presented in this thesis mark the first time that molecular simulation has been used to obtain phase diagrams describing all types of equilibria between vapor, liquid, and solid phases. The Gibbs-Duhem integration approach has two main advantages compared to the equation of state approach for calculating phase equilibria: (1) predictions made by an equation of state are subject to spurious error as a result of approximations, while the Gibbs-Duhem integration predictions are based on simulation results, which are exact for the particular model being studied, and (2) equations of state do not allow for solid phase formation but a solid phase with any crystalline structure can be readily incorporated into the Gibbs-Duhem integration scheme.

The limitations of this approach should be noted as well. First, the Gibbs-Duhem integration procedure, as formulated here, cannot be used to evaluate critical phenomena, as was demonstrated in Chapter 4. Second, it is not always trivial to obtain

an initial condition to begin the integration. For example, there is presently no straightforward way to calculate liquid-liquid and solid-solid coexistence lines that are not connected (by way of interference with a vapor-liquid or solid-liquid coexistence region) to a pure component coexistence point. In this case, one would have to conduct several exploratory Gibbs-Duhem integrations along constant pressure and constant temperature paths to obtain an initial coexistence point, as illustrated in Figure 6.1. Here, at pressure, P_c, a closed loop of liquid-liquid immiscibility is detached from the vapor-liquid coexistence region. To obtain an initial coexistence condition for the liquid-liquid immiscibility curve at P_c , one could start by conducting a constant pressure Gibbs-Duhem integration at P_a. The liquid-liquid coexistence points determined at T_b become the initial coexistence condition for a constant temperature (T_b) Gibbs-Duhem integration of the liquid-liquid immiscibility curve from P_a to P_c. Using the liquid-liquid coexistence point at P_c as the initial coexistence condition the liquid-liquid coexistence curve at P_c can now be calculated.

Calculating phase diagrams via Gibbs-Duhem integration affords the possibility of classifying phase diagrams based not only on fluid phase behavior, as was done by Van Konynenburg and Scott[1,2], but also on solid phase behavior. There are two questions regarding the fluid phase equilibria classifications based on equations of state that the Gibbs-Duhem integration method would be particularly well-suited to address. The first question is 'do the Type I and Type V classes of phase behavior truly exist for mixtures of light alkanes, as is commonly believed, or are they really just Type II and Type IV classes, respectively, where the low temperature liquid-

liquid immiscibility has been covered up by the solid phase equilibria?'.[3,4] This can be answered by applying the Gibbs-Duhem integration method, in a manner similar to that demonstrated in Chapter 3, to calculate liquid-liquid equilibria that is metastable with respect to equilibria involving solid phases. The second question, 'can the Type VI (closed-loop liquid-lquid immiscibility) class exist for a mixture of spherical molecules without directional interactions?' has been the subject of a spirited debate in the literature[5–9]. Type VI phase behavior has been discovered in equation of state models for mixtures of spherically symmetric molecules, however, many skeptics think that this finding is a mathematical artifact brought about by assumptions made in the development of the equations of state. As we mentioned above, using a Gibbs-Duhem approach based on simulations would provide exact results and could shed some light onto this matter.

In Chapter 1, we stated that our primary motivation for calculating solid-fluid phase equilibria is that we eventually want to compute phase equilibria for a ternary mixture containing a pair of model diastereomers and a solvent. We have successfully accomplished the first step towards this goal by developing a molecular simulation tool to calculate solid-fluid equilibria for binary mixtures of Lennard-Jones molecules. The next steps in this research program are to extend the method to ternary mixtures of spherical molecules (adapting the code to allow for three components should be a trivial undertaking) and to extend the method to include non-spherical molecules. The latter step promises to be a challenge but some recent developments in molecular simulation methods for solid-liquid phase equilibria of

chain-like molecules should prove useful in this case. Malanoski and Monson[10] recently determined solid-liquid coexistence for a single component system containing flexible hard sphere chains of lengths n = 3,4,5,6,7, and 8 using thermodynamic integration techniques to calculate the free energy of each phase and determine coexistence points. They later extended their method to a binary mixture of chains of lengths n = 7 and 8[11]. Polson and Frenkel[12] determined solid-liquid coexistence for flexible Lennard-Jones chains of lengths n = 6 and 10 using the Gibbs-Duhem integration method. Their initial coexistence condition came from thermodynamic integration techniques that were similar to Malanoski and Monson's. These methods could be extended to calculate solid-liquid coexistence for a model diastereomer molecule, which could then serve as an initial coexistence condition for calculating a solid-liquid coexistence curve for a mixture of diastereomers and solvent.

6.1 References

[1] R. L. Scott and P. H. van Konynenburg, Discussions of the Faraday Society, **49**, 87–97 (1970).

[2] P. H. van Konynenburg and R. L. Scott, Philosophical Transactions of the Royal Society of London A., **298**, 495–540 (1980).

[3] R. L. Scott, Accounts of Chemical Research, **20**, 97–107 (1987).

[4] C. J. Peters. personal communication.

[5] L. Z. Boshkov and L. V. Yelash, Fluid Phase Equilibria, **141**, 105–112 (1997).

[6] L. V. Yelash and T. Kraska, Physical Chemistry Chemical Physics, **1**, 307–311 (1999).

[7] M. N. Garcia-Lisbona, A. Galindo, G. Jackson, and A. N. Burgess, Molecular Physics, **93**(1), 57–71 (1998).

[8] G. Jackson, Molecular Physics, **72**(6), 1365–1385 (1991).

[9] L. A. Davies, G. Jackson, and L. F. Rull, Physical Review E, **61**(3), 2245–2256 (2000).

[10] A. P. Malanoski and P. A. Monson, Journal of Chemical Physics, **110**(1), 664–675 (1999).

[11] A. P. Malanoski and P. A. Monson, Journal of Chemical Physics, **112**(6), 2870–2877 (2000).

[12] J. M. Polson and D. Frenkel, Journal of Chemical Physics, **109**(1), 318–328 (1998).

6.2 Figures

Page

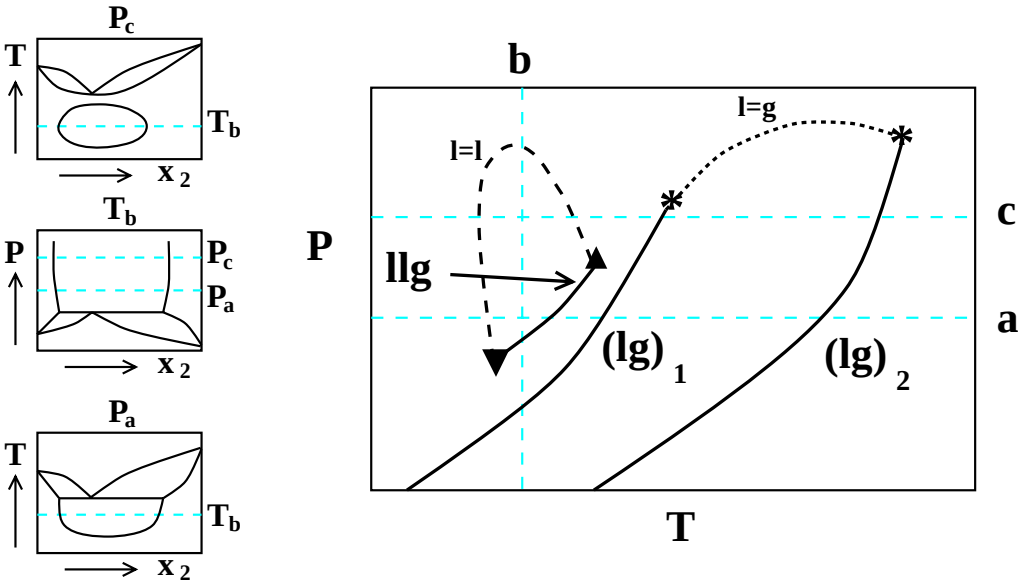

Figure 6.1: Schematic diagram illustrating a series of constant pressure and constant temperature paths that could be used to locate closed-loop liquid-liquid immiscibility. Solid lines represent pure component liquid-vapor coexistence lines (lg) and the liquid-liquid-vapor coexistence line (llg). Asterisks denote the pure component liquid-vapor critical points. The black, short-dashed line represents the locus of liquid-vapor critical points and the black, long-dashed line represents the the locus of liquid-liquid critical points for the mixture. The gray dashed-lines indicate the constant temperature and constant pressure paths.

Appendices

APPENDIX A

SOLID-LIQUID PHASE EQUILIBRIUM FOR BINARY LENNARD-JONES MIXTURES

A.1 Introduction

Successful design of melt crystallization processes depends upon knowledge of the equilibrium solid-liquid phase behavior of the mixture to be separated. Although mixture solid-liquid equilibrium can, in principle, be measured at any condition of interest, it would be useful to develop a theory that predicts mixture phase behavior based solely upon knowledge of the components' molecular architecture and intermolecular forces. Such a theory should be able to predict the six types of binary solid-liquid phase diagrams introduced in 1977 by Matsuoka[1], a classification based on the analysis of nearly 1500 binary organic systems. The six types of binary solid-liquid phase diagrams are shown in Figure A.1; they include: solid solutions, azeotropes, eutectics with partial solid phase immiscibility, eutectics with complete solid phase immiscibility, peritectics with eutectics, and molecular compounds.

Our intent in this paper is to explore the range of Lennard-Jones parameters

that will yield the phase diagrams characteristic of solid solutions, azeotropes, and eutectics with partial solid phase immiscibility. The solid phase in all of these phase diagrams is a substitutionally disordered solid solution, i.e., both species pack with the same crystalline structure and can substitute for one another in any order on the lattice. In future work we will examine the effect of Lennard-Jones parameters on solid-liquid equilibrium for cases in which the solid phase is an ordered crystal and the associated phase diagram contains either a peritectic with eutectic or a molecular compound.

Several workers have developed molecularly-based theories of solid-liquid phase equilibria using simple models, such as the hard-sphere model or the Lennard-Jones 12-6 potential model, in an effort to understand how molecular size differences and interactions influence the phase behavior of a mixture. Cottin and Monson[2] developed a cell theory for binary solid solutions based on the early cell model of Lennard-Jones and Devonshire for single component solids. In this model, each molecule is assumed to move within the cage formed by its nearest neighbors. A cell is characterized by the identity of its central molecule and by the composition and geometrical arrangement of its nearest neighbors. Assuming that the product of the cell partition functions is equal to the configurational partition function, the solid solution free energy can be determined as a function of composition. The liquid solution free energy can be obtained from an equation of state. Once the free energies of both phases are obtained, the convex envelope construction method is used to locate the solid-liquid phase envelope.

The cell theory has been applied to phase diagram calculations for both hard-sphere and Lennard-Jones binary mixtures. Cottin and Monson calculated phase diagrams for binary hard-sphere liquid mixtures coexisting with substitutionally disordered fcc[2] and several ordered[3] binary hard-sphere solids. Cottin and coworkers[4,5] showed that the cell theory is capable of predicting five of the six solid-liquid phase diagrams displayed in Figure A.1. The only type of phase diagram that the cell theory with the hard-sphere potential model does not predict is that for molecular compound formation. The fact that the hard-sphere model can generate five of the six known types of solid-liquid phase diagrams indicates that packing and molecular size differences, which are the major consideration in a hard-sphere mixture, are the dominant factors in determining the shape of solid-liquid phase diagrams in real systems.

Cottin and Monson also calculated phase diagrams for binary Lennard-Jones liquid mixtures[6] coexisting with substitutionally disordered fcc binary Lennard-Jones solids. They calculated phase diagrams for argon-methane, argon-krypton, and krypton-methane, and compared them with experimental results. They found that the shape of the phase diagram predicted by cell theory is similar to the shape of the phase diagram observed experimentally.

Another approach to calculating solid-liquid equilibria is density functional theory. In density functional theory, a crystalline solid is treated as if it were an inhomogeneous liquid[7]. There seem to be two main approaches to developing analytical expressions for the free energy of the solid: truncated perturbation expansions in

which the excess free energy is expanded about a uniform reference liquid[8,9], and non-perturbative effective liquid approximations in which the excess free energy of the non-uniform liquid (crystalline solid) is approximated by the excess free energy of a uniform liquid evaluated at an effective density[10,11]. The liquid solution free energy can be obtained from an equation of state. As in the cell theory, once the free energies of both phases are obtained, the convex envelope construction method is used to locate the solid-liquid phase envelope.

Density functional theory has been applied to the calculation of phase diagrams for hard-sphere and Lennard-Jones binary mixtures. Rick and Haymet[8,9], Denton and Ashcroft[11], and Zeng and Oxtoby[10] used density functional theory to calculate phase diagrams for binary hard-sphere liquid mixtures coexisting with substitutionally disordered fcc hard-sphere solid solutions. Rick and Haymet, and Denton and Ashcroft also investigated the range of stability for several ordered binary hard-sphere solid solutions coexisting with a liquid, where the composition of each phase was fixed at $x = 0.5$. The non-perturbative theories (Denton and Ashcroft, Zeng and Oxtoby) give more accurate results than the perturbative theories (Rick and Haymet). As was the case with the cell theory, the density functional theory applied to binary hard sphere mixtures shows that size differences alone can determine the shape of the phase diagram. Density functional theory is not as successful as cell theory at predicting phase behavior in binary Lennard-Jones systems. Rick and Haymet used density functional theory to calculate solid-liquid phase diagrams for binary Lennard-Jones mixtures of argon-methane, argon-krypton, and krypton-methane, and compared

them with experimental results. Of the three systems, they were only able to predict the shape of the krypton-methane phase diagram correctly.

It is common practice to test molecularly-based theories, such as cell theory and density functional theory, with molecular simulation data. Since molecular simulations are exact calculations for a given molecular model, they provide a direct test of theoretical approximations. The approach taken in most simulations of solid-liquid phase equilibrium involves calculating the free energy of each phase separately and then determining the coexistence curve using the convex envelope construction method. Since the free energy cannot be directly measured in a simulation, thermodynamic integration techniques are often used. In thermodynamic integration, the free energy of a phase is calculated by numerically integrating along a reversible path from a reference state to the desired state. This integration requires simulation data at many intermediate state points. The liquid phase free energy can be calculated by thermodynamic integration from the ideal gas reference state. The solid phase free energy cannot be determined by thermodynamic integration from the ideal gas state because this path crosses a first-order transition, where appreciable hysteresis may occur, and is irreversible[12]. There are, however, other free energy reference states that can be used. Frenkel and Ladd[13] have used the Einstein crystal as a reference state to calculate the free energy of a hard-sphere solid via thermodynamic integration. An Einstein crystal consists of regularly-spaced molecules that are held in place on a lattice by the repulsive forces between their nearest neighbors[14]. Eldridge *et al.*[15–17] have used Frenkel and Ladd's Einstein crystal integration

method to calculate the free energies of binary hard-sphere solids with ordered phases such as AB, AB_2, and AB_{13}. Frenkel and Ladd's integration method is not well-suited for substitutionally disordered solid phases, i.e., solid solutions, because there are relatively small free energy differences between the real mixture and the Einstein reference mixture[18].

Other thermodynamic integration approaches have been taken for solid-liquid equilibrium calculations when the solid is a disordered crystal. Kranendonk and Frenkel[18,19] developed two methods for calculating the free energies of disordered fcc binary hard-sphere solid solutions using the free energy of a pure hard sphere fcc solid, as calculated by Frenkel and Ladd, as the reference state. In one method, they construct a reversible path from the pure hard-sphere fcc solid reference by fixing the density and diameter ratio and varying the composition from unity to the composition of the hard-sphere mixture using particle insertion techniques. In a second method, they construct a reversible path by fixing the density and composition and varying the diameter ratio from unity to the diameter ratio of the hard-sphere mixture. Kofke[20] introduced another thermodynamic integration method for calculating solid-liquid coexistence for cases in which the solid is a disordered binary hard-sphere fcc solid solution. This method utilizes semigrand canonical ensemble simulations[21,22] in which the number of molecules, temperature, pressure, and species fugacity fractions are fixed and the density and composition are allowed to vary. Kofke used semigrand canonical simulations to generate composition versus fugacity fraction data for several solid phase isotherms. Using this data and the

chemical potential of one of the pure species as a reference (taken from the data of Hoover and Ree[23]), Kofke integrated the semigrand canonical fundamental equation to calculate the species chemical potential as a function of fugacity fraction for each isotherm. The chemical potential versus fugacity fraction curve for the liquid phase was obtained from a hard-sphere equation of state. Since the species chemical potentials and fugacity fractions must be equal at equilibrium, phase coexistence could be found for each isotherm by determining the intersection of the chemical potential versus fugacity fraction curves.

Using cell theory and the molecular simulation techniques described above, several workers have determined the range of size ratios in binary hard-sphere mixtures that yield the various types of phase diagrams. Table A.1 lists hard-sphere diameter ratios that produce solid-liquid phase diagrams of the solid solution, azeotrope and eutectic types, along with the diameter ratio boundaries between these three types. These substitutionally disordered solids are generally stable with respect to phase separation at diameter ratios greater than 0.85. For lower diameter ratios substitutionally ordered solids are stable. Table A.2 gives the range of diameter ratios over which the structures, AB, AB_2 and AB_{13} are stable in hard-sphere mixtures.

To our knowledge, the only simulation study of solid-liquid equilibrium in binary Lennard-Jones mixtures is that of Vlot *et al.*[24] who calculated solid-fluid phase diagrams for symmetric (equal diameters, $\sigma_{11} = \sigma_{22}$; equal attractions, $\epsilon_{11} = \epsilon_{22}$) binary Lennard-Jones mixtures using a combination of molecular simulation and theory. Cross interactions between the components were defined as $\epsilon_{12} = e\sqrt{\epsilon_{11}\epsilon_{22}}$

and $\sigma_{12} = s\,(\sigma_{11} + \sigma_{22})/2$ where the parameters e and s represent deviations from the Lorentz-Berthelot mixing rules[25]. They used a variety of thermodynamic integration and particle insertion techniques to calculate free energies. The solid phase free energy was calculated using the Einstein crystal reference state[26], the liquid phase free energy was calculated using the *fg* sampling method[27,28], and the gas phase free energy was calculated using Widom's test particle insertion[29]. The excess free energy versus composition data for all of the phases was fit with a two-parameter Redlich-Kister polynomial[30]. The convex envelope construction method was used to determine the temperature versus composition phase diagrams.

In the simulation techniques that we have discussed thus far, phase equilibrium is calculated indirectly in that many simulations must be conducted at uninteresting state points in order to get one coexistence point. Prior to 1987, direct simulation of phase equilibrium was considered to be too computationally expensive because the presence of an interface requires the system to contain an unusually large number of particles. Then in 1987, Panagiotopoulos[31] introduced the Gibbs ensemble method for direct simulation of phase coexistence in single component fluids and in 1988, extended the method to phase coexistence in multicomponent mixtures[32]. In this method the coexisting phases are simulated independently, so as to remove the interface, yet are essentially coupled together in order to satisfy the criteria of phase equilibrium: equal temperatures, pressures, and species chemical potentials. These conditions are satisfied by simulating each phase at the same temperature and pressure, and performing particle exchanges between the phases to maintain chem-

ical potential equality. The Gibbs ensemble method has been widely used to study vapor-liquid and liquid-liquid phase equilibria. However, because a large number of successful transfers of particles between each phase are required for chemical potential equilibration[33], the Gibbs ensemble method is not an efficient method for studying solid-liquid phase equilibrium.

Inspired by the Gibbs ensemble method, Kofke[34,35] introduced the Gibbs-Duhem integration technique for direct simulation of phase equilibria. As in the Gibbs ensemble method, two or more coexisting phases are simulated independently at the same temperature and pressure. However, instead of using particle exchanges, Kofke maintains chemical potential equality in each phase by integrating along the Clapeyron equation for coexistence during the simulations. Eliminating the need for particle transfers between phases makes the Gibbs-Duhem integration method well-suited for calculating phase equilibrium for cases in which one of the phases is a solid. The Gibbs-Duhem integration simulation approach has been used to tackle a number of multicomponent, multiphase equilibrium problems, including: finding the triple point of the Lennard-Jones fluid[36,37], determining the melting line of C_{60}[38], and calculating phase diagrams for colloids in polymer solution[39] and for polymer-induced protein precipitation[40]. In each case the appropriate Clapeyron differential equation is derived and an initial coexistence condition is obtained from theory, experimental data, or a Gibbs ensemble Monte Carlo simulation.

In this paper, we combine Kofke's Gibbs-Duhem integration technique with semigrand canonical Monte Carlo simulations to calculate solid-liquid phase diagrams

for binary Lennard-Jones mixtures. To our knowledge, these are the first *direct* simulations of solid-liquid equilibrium in binary Lennard-Jones mixtures.

We calculate solid-liquid phase diagrams for binary Lennard-Jones mixtures with diameter ratios ranging from 0.85 to 1 and attractive well-depth ratios ranging from 0.45 to 1.6, at a reduced pressure equivalent to 1 atm. We determine the cross-species interaction parameters by using the Lorentz-Berthelot[25] mixing rules. The solid-liquid equilibrium line is calculated for each mixture by numerically integrating the Clapeyron differential equation for binary mixture phase equilibria at constant pressure. The initial condition for the integration is that of a single component Lennard-Jones liquid in equilibrium with a single component Lennard-Jones fcc crystalline solid at atmospheric pressure. The coexistence properties at subsequent integration points are determined by semigrand canonical Monte Carlo simulations (constant temperature, pressure, total number of molecules, and fugacity fraction) of the liquid and the fcc solid phases.

Highlights of our simulation results are the following. In a test of the cell theory predictions for model argon-methane, argon-krypton, and krypton-methane systems, we found that cell theory can qualitatively predict the shape of phase diagrams for Lennard-Jones mixtures. Comparison of our simulation results for the argon-krypton Lennard-Jones system with hard sphere results at the same diameter ratio indicated that the presence of attractive interactions can change the type of phase diagram observed from azeotrope (hard sphere) to solid solution (Lennard-Jones). This suggested that attractive interactions are an important factor in determining the

type of solid-liquid phase behavior observed and prompted a more thorough investigation of the effect of variations in well-depth ratio on solid-liquid phase diagrams. We simulated 56 binary Lennard-Jones mixtures over the range of parameters listed above to determine how solid-liquid phase diagrams change as a function of diameter ratio and well-depth ratio. We found that for well-depth ratios of unity (equal attractions among species) phase behavior indicative of azeotropes and eutectics is observed for diameter ratios ranging from 0.85 to 1. We then varied the well-depth ratio of the mixtures at several constant diameter ratios and observed transitions from azeotrope to solid solution, from azeotrope to eutectic, and from solid solution to simple peritectic. Using our simulation results, we are able to map out the boundaries separating regimes of solid solution, azeotrope and eutectic solid-liquid phase behavior in the space spanned by the Lennard-Jones diameter ratio and well-depth ratios.

The remainder of the paper is organized as follows. Section A.2 reviews the Gibbs-Duhem integration method and gives details of how we applied the technique to our study of binary solid-liquid mixture equilibrium. In Section A.3, the simulation results are presented and discussed. A brief summary and further discussion are given in Section A.4.

A.2 Method

In this section we describe how we calculated solid-liquid equilibria in binary Lennard-Jones mixtures using the Gibbs-Duhem integration method. We begin by

presenting a brief overview of the Gibbs-Duhem integration method. We then discuss our procedures for determining an initial coexistence condition and integrating the Clapeyron equation. Finally, we describe the details of the semigrand ensemble simulations used throughout the integration procedure.

In the Gibbs-Duhem integration method[34,35,41], phase coexistence is determined by numerically integrating the Clapeyron differential equation appropriate to the system of interest. Clapeyron equations describe how field variables (variables that must be equal among coexisting phases) change along the phase equilibrium line. The Clapeyron equation for equilibrium between a binary solid mixture and a binary liquid mixture at constant pressure is[35]

$$\frac{d\beta}{d\xi_2} = \frac{(x_2^l - x_2^s)}{\xi_2(1-\xi_2)(h^l - h^s)}, \tag{A.1}$$

where β is the reciprocal temperature, $1/kT$, with k the Boltzmann constant and T the absolute temperature, ξ_2 is the fugacity fraction of species 2, $\xi_2 \equiv \hat{f}_2 / \sum \hat{f}_i$, with $\hat{f}_i$, the fugacity of species i in solution, x_2 is the mole fraction of species 2, h is the molar enthalpy, and the superscripts l and s denote the liquid and solid phases, respectively. The right-hand side of Eq. (A.1) can be integrated numerically to find an equation for β as a function of ξ_2 if we have an initial condition describing the temperature, fugacity fraction, enthalpies and compositions at one coexistence point.

A.2.1 Initial condition

We need an initial coexistence condition before we can begin a Gibbs-Duhem integration calculation of solid-liquid equilibrium in a binary mixture. A good choice is the solid-liquid equilibrium condition for either of the pure components[35]. Since the slope of the integrand in Eq. (A.1) is undefined for pure components ($\xi_2 = 0, x_2 = 0$ and $\xi_2 = 1, x_2 = 1$), we instead estimate it using the limiting case of infinite dilution. Here we follow Mehta and Kofke[42], who used the infinite dilution case to start their Gibbs-Duhem integration calculations of vapor-liquid equilibria in binary mixtures.

Consider the case of pure species 1 solid-liquid coexistence. We can estimate $(d\beta/d\xi_2)_{x_2=0}$ by supposing that the real mixture displays ideal solution behavior at the limit of infinite dilution of species 2. With this assumption, the abundant component (species 1) in the ideal solution follows the Lewis Randall rule

$$\hat{f}_1 = x_1 f_1, \tag{A.2}$$

while the dilute component (species 2) obeys Henry's law

$$\hat{f}_2 = x_2 H_2, \tag{A.3}$$

where $\hat{f}_1$ and $\hat{f}_2$ are fugacities of species 1 and 2 in solution, f_1 is the fugacity of pure component 1 at the temperature and pressure of the mixture, and H_2 is the Henry's law constant for species 2. Letting $x_1 \to 1$ and $x_2 \to 0$, the fugacity fraction

of species 2 becomes, $\xi_2 = x_2 H_2 / f_1$. After making these substitutions into Eq. (A.1) we get

$$\left.\frac{d\beta}{d\xi_2}\right|_{x_2=0} = \frac{\hat{f}_2/H_2^l - \hat{f}_2/H_2^s}{(x_2 H_2/f_1)(1 - x_2 H_2/f_1)(h^l - h^s)}, \tag{A.4}$$

which, after simplification, leads to

$$\left.\frac{d\beta}{d\xi_2}\right|_{x_2=0} = \frac{f_1(1/H_2^l - 1/H_2^s)}{(h^l - h^s)}. \tag{A.5}$$

This gives us an estimate of the integrand at the initial condition of $\xi_2 = 0$ and melting temperature, T_1, of pure species 1.

We can calculate all of the quantities on the right-hand side of Eq. (A.5) with an NPT simulation of pure species 1. The molar enthalpy of each phase is $h = \langle u + Pv \rangle_{\mathrm{NPT}}$, where u is the configurational energy of the system and the brackets, $\langle\rangle_{\mathrm{NPT}}$, denote an NPT ensemble average. The quantity f_1/H_2 is given by[42]

$$\frac{f_1}{H_2} = \langle \exp(-\beta \Delta u_{1\to 2}) \rangle_{\mathrm{NPT}} \tag{A.6}$$

where $\Delta u_{1\to 2}$ is the exchange energy associated with switching a particle from species 1 to 2. The exchange energy, $\Delta u_{1\to 2}$, can be obtained by conducting trial identity switches during the simulation. A trial identity switch consists of randomly selecting a particle and calculating the energy that would result if we were to switch the particle from species 1 to species 2. This is done without actually changing the particle's identity.

A.2.2 Integration

Once we have an initial coexistence condition, we can begin the Gibbs-Duhem integration series using a predictor-corrector algorithm to integrate Eq. (A.1) over the entire range of fugacity fractions, $\xi_2 = 0$ to $\xi_2 = 1$. Stepping to the next fugacity fraction, $(\xi_2)_1$, from the initial data point, the next reciprocal temperature, β_1, can be estimated using the trapezoid-rule predictor formula

$$\beta_1^{(0)} = \beta_0 + [(\xi_2)_1 - (\xi_2)_0]\, F(\beta_0, (\xi_2)_0), \tag{A.7}$$

where the "0" superscript indicates that $\beta_1^{(0)}$ (a predicted value) is our zeroth iteration attempt at finding the reciprocal temperature β_1, $(\xi_2)_0$ and β_0 are the initial fugacity fraction and reciprocal temperature, respectively, and F is the right-hand side of Eq. (A.1) evaluated at the initial condition. Once $\beta_1^{(0)}$ is estimated at the given fugacity fraction $(\xi_2)_1$, two semigrand canonical (NPTξ_2) Monte Carlo simulations (one for the liquid phase and one for the solid phase) are conducted in order to equilibrate the configurations and to calculate the enthalpies and mole fractions at the new state point. (Details of the NPTξ_2 simulations will be given in Section A.2.3.)

After each phase is equilibrated we refine our estimate for β_1 with the trapezoid-rule corrector

$$\beta_1^{(1)} = \beta_0 + \frac{[(\xi_2)_1 - (\xi_2)_0]}{2}\left[F_1^{(0)}(\beta_1^{(0)}, (\xi_2)_1) + F_0(\beta_0, (\xi_2)_0)\right], \tag{A.8}$$

where the subscripts "0" and "1" denote the initial and current conditions, respec-

tively and $F_1^{(0)}$ is calculated from simulation averages of the enthalpies and mole fractions at $\beta_1^{(0)}$ and $(\xi_2)_1$. After the reciprocal temperature is corrected, two NPTξ_2 simulations, one for each phase, are started simultaneously and running averages of the compositions and enthalpies are calculated. Each simulation is paused periodically and the running averages from the simulations to that point are used to calculate $F_1^{(i)}$ and correct the reciprocal temperature to obtain the $(i+1)^{\text{th}}$ estimate, $\beta_1^{(i+1)}$. This pattern continues until the temperature estimate stops varying within a specified tolerance. A final production segment of simulations are run at the temperature determined from the corrector segment to obtain the final average enthalpies and mole fractions for the coexistence point.

After β_1 converges and the production runs are completed, the fugacity fraction is incremented to the next state point and the predictor-corrector algorithm described above is repeated. Higher order predictor-corrector equations are used as we obtain more coexistence points. The midpoint predictor-corrector[35] is used once two state points are known,

$$\beta_{n+1}^{(0)} = \beta_{n-1} + 2[(\xi_2)_{n+1} - (\xi_2)_n]\, F_n(\beta_n, (\xi_2)_n), \tag{A.9}$$

$$\beta_{n+1}^{(i+1)} = \beta_{n-1} + \frac{[(\xi_2)_{n+1} - (\xi_2)_n]}{3}\left[F_{n+1}^{(i)} + 4F_n + F_{n-1}\right], \tag{A.10}$$

and the modified Adams predictor-corrector[43] is used once three or more state

points are known,

$$\beta_{n+1}^{(0)} = \beta_n + \frac{[(\xi_2)_{n+1} - (\xi_2)_n]}{24}[55F_n - 59F_{n-1} + 37F_{n-2} - 9F_{n-3}], \tag{A.11}$$

$$\beta_{n+1}^{(i+1)} = \beta_n + \frac{[(\xi_2)_{n+1} - (\xi_2)_n]}{24}\left[9F_{n+1}^{(i)} + 19F_n - 5F_{n-1} + F_{n-2}\right]. \tag{A.12}$$

In these sets of equations the predictor is listed first. The subscripts denote the coexistence conditions determined from simulations with $(n+1)$ being the current simulation and the superscripts denote the iterations of the corrector for the current coexistence point. We mapped the entire temperature versus composition phase diagram by repeating the predictor-corrector algorithm from $\xi_2 = 0$ to $\xi_2 = 1$.

A.2.3 Simulations

The enthalpies and mole fractions needed as input to the integration of Eq. (A.1) are obtained from semigrand canonical (constant NPTξ_2) Monte Carlo computer simulations[22]. In this work, all simulations were run with a system size of 500 particles at a reduced pressure equivalent to 1 atm. The temperature and fugacity fraction were varied according to the values specified by the Gibbs-Duhem integration predictor-corrector. There are three types of Monte Carlo trial moves in semigrand canonical simulations: particle displacements, volume change moves, and particle identity exchanges. The particle displacements and volume change moves are conducted just as they are in a standard NPT simulation[44]. In the particle identity exchange moves, a particle is selected at random and given a trial species identity

switch. The identity switch is accepted according to the ratio of the species fugacity fractions, ξ_1 and ξ_2. The overall acceptance probability[42] for the moves in the NPTξ_2 ensemble is $\min[1, \exp(\Lambda)]$ where

$$\Lambda = -\beta(U^{\text{trial}} - U^{\text{old}}) - \beta P(V^{\text{trial}} - V^{\text{old}}) + N \ln \frac{V^{\text{trial}}}{V^{\text{old}}} + m \ln \frac{\xi_2}{1 - \xi_2}. \qquad \text{(A.13)}$$

In Eq. (A.13), U^{trial} and U^{old}, and V^{trial} and V^{old}, are the configurational energies and volumes of the trial and existing states, respectively, $m = +1$ if the trial identity switch is from species 1 to 2, and $m = -1$ if the trial identity switch is from species 2 to 1. In NPTξ_2 simulations the choice of the type of Monte Carlo move is made randomly but weighted such that the ratio of attempted moves is 1 volume change to N particle displacements to N identity switches. The length of the simulation is given in cycles, where one cycle represents either N displacement attempts, N identity switch attempts, or 1 volume change attempt. In our work, a typical NPTξ_2 simulation is equilibrated for 15 000 cycles and then followed by a production run of 3000 cycles to compute the average enthalpy and mole fraction. The only difference between solid and liquid phase simulations is that to maintain an fcc crystalline structure in the solid phase simulations we impose a single occupancy constraint[20,45] on the trial displacements of particles in the solid, *i.e.,* any displacements that put the particle outside its lattice cell are rejected.

Other details of the NPTξ_2 simulations are as follows. The simulation volume is a cubic box with periodic boundary conditions. The particles interact via the

Lennard-Jones potential model

$$u_{ij}(r) = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{r}\right)^{12} - \left(\frac{\sigma_{ij}}{r}\right)^{6}\right], \tag{A.14}$$

where u_{ij} is the potential energy of interaction between particles i and j, r is the distance between particles i and j, ϵ_{ij} is the Lennard-Jones attractive well-depth, and σ_{ij} is the Lennard-Jones diameter. We determine the cross-species interaction parameters $(\sigma_{12}, \epsilon_{12})$ by using the Lorentz-Berthelot[25] mixing rules $\sigma_{12} = (\sigma_{11} + \sigma_{22})/2$ and $\epsilon_{12} = \sqrt{\epsilon_{11}\epsilon_{22}}$. The potential interactions are truncated at a cutoff radius of half the box length. To compensate for this truncation, a long range correction is applied to the energy and virial calculations during the simulation by assuming a uniform density distribution beyond the cutoff radius[44].

A.3 Results and Discussion

In this section we describe the results of our Gibbs-Duhem integration calculation of solid-liquid phase diagrams for binary Lennard-Jones systems. We begin by comparing our simulation results with cell theory results and with experimental data. Since our molecular simulations are exact calculations for the Lennard-Jones potential model we can compare the simulation data to experimental results to test the ability of the Lennard-Jones model to mimic behavior in real systems. We can also compare the simulation data to the cell theory results to test the cell theory's ability to predict the behavior of the Lennard-Jones system. We then examine how varia-

tions in the ratios of Lennard-Jones diameters (σ_{11}/σ_{22}) and well-depths ($\epsilon_{11}/\epsilon_{22}$) influence the types of phase diagrams that will be observed.

A.3.1 Comparison with molecular theory and experiment

Following Cottin and Monson[6], we calculated temperature-composition diagrams for the model mixtures: argon-methane, argon-krypton, and krypton-methane, at $P = 1$ atm. These phase diagrams were then compared with Cottin and Monson's cell theory predictions and with experimental data.

We used the same Lennard-Jones parameters as Cottin and Monson; the diameter (σ_{ii}) and well-depth (ϵ_{ii}) for argon, krypton, and methane are listed in Table A.3. We worked in reduced units $T^* = kT/\epsilon_{11}$ and $P^* = P\sigma_{11}^3/\epsilon_{11}$. For $P = 1$ atm, the corresponding reduced pressures are $P^* = 0.00193$ (argon), $P^* = 0.00175$ (krypton), and $P^* = 0.00243$ (methane).

The initial condition for the Gibbs-Duhem integration calculations came from the simulation data of Agrawal and Kofke[37] who calculated the solid-liquid equilibrium line for a single component Lennard-Jones system. They obtained a reduced melting temperature of $T^* = 0.687$ for the range of reduced pressures, $P^* = 0.0008$ to 0.008. In our calculations, we started with an initial condition of $T_1^* = kT_1/\epsilon_{11} = 0.687$, where T_1^* and T_1 are the reduced and absolute melting temperatures of component 1, respectively. We estimated the uncertainty in the simulation data by performing three Gibbs-Duhem integration simulations for each mixture and calculating the standard deviation of the temperature and mole frac-

tions of the liquid and solid phases at each coexistence point. In all three mixtures, the standard deviation in coexistence temperature, kT/ϵ_{11}, is less than 0.2% and in mole fraction, x_2, is less than 0.6%.

Figure A.2 shows the temperature-composition phase diagram for the argon-methane system obtained using Gibbs-Duhem integration starting from pure methane solid-liquid coexistence, along with the cell theory prediction of Cottin and Monson and the experimental results of van't Zelfde *et al.*[46]; the cell theory and experimental data have been reduced with the Lennard-Jones parameters given in Table A.3. The diameter ratio is $\sigma_{Ar}/\sigma_{CH_4} = 0.9$ and the well-depth ratio is $\epsilon_{Ar}/\epsilon_{CH_4} = 0.88$. We obtain a minimum azeotrope similar to the experimental data and cell theory prediction. A glance at the comparison of the ability of simulations and theory to predict experimental behavior would, at first, lead one to think that the theory is more accurate than the simulations. However, as Cottin[4] points out, direct comparison of theoretical predictions with experimental data is difficult because inadequacies in the potential model coupled with simplifying approximations made in the theory could lead to error cancellation. This cancellation effect is illustrated in Figure A.2 where the Lennard-Jones model (simulation) over predicts melting temperatures for real systems (experiments), and cell theory under predicts melting temperatures for the Lennard-Jones model. There is a good agreement in the predicted azeotrope composition between cell theory and the Lennard-Jones model, although neither predicts the azeotrope composition found in experiments. After making these comparisons, it is also useful to compare the phase diagram

of this Lennard-Jones mixture with a hard-sphere mixture of the same diameter ratio to determine whether attractive interactions are contributing to the type of phase behavior observed. In this case, a hard-sphere mixture with the diameter ratio $\sigma_{Ar}/\sigma_{CH_4} = 0.9$ would also display azeotrope phase behavior.

Figure A.3 shows the temperature-composition phase diagram of the krypton-methane system obtained using Gibbs-Duhem integration starting from pure methane solid-liquid coexistence, along with the cell theory prediction and the experimental results of Veith and Schroder[47]. The diameter ratio is $\sigma_{Kr}/\sigma_{CH_4} = 0.96$ and the well-depth ratio is $\epsilon_{Kr}/\epsilon_{CH_4} = 1.2$. We obtain a spindle-shaped phase diagram very similar to the experimental data and cell theory prediction. A hard-sphere mixture with the diameter ratio $\sigma_{Kr}/\sigma_{CH_4} = 0.96$ would also display solid solution phase behavior.

Figure A.4 shows the temperature-composition phase diagram for the argon-krypton system obtained using Gibbs-Duhem integration starting from pure argon solid-liquid coexistence, along with the cell theory prediction and the experimental results of Heastie[48]. Here the diameter ratio is $\sigma_{Ar}/\sigma_{Kr} = 0.93$ and the well-depth ratio is $\epsilon_{Ar}/\epsilon_{Kr} = 0.74$. We obtain a spindle-shaped phase diagram similar to the experimental data and the cell theory prediction. The simulation matches the shape of the experimental data, especially in the argon-rich region where the temperature-composition line tends to level off. A hard-sphere mixture with a diameter ratio $\sigma_{Ar}/\sigma_{Kr} = 0.93$ would not exhibit solid solution behavior but instead would have a minimum azeotrope. This suggests that the intermolecular attractions between

molecules can be a significant factor in determining the type of solid-liquid phase diagram observed.

A.3.2 Influence of attractions on phase behavior

As we have discussed in Section A.1, five of the six different types of solid-liquid phase diagrams for binary mixtures can be predicted by using a hard-sphere model with an appropriate diameter ratio. Of the three Lennard-Jones mixtures discussed thus far, we have seen one instance (argon-krypton) where the difference in attractive interactions was large enough to change the type of phase diagram observed from azeotrope (hard sphere) to solid solution (Lennard-Jones). In order to get a better idea of what role attractive interactions play in solid-liquid mixture phase equilibria, we performed a systematic investigation of how solid-liquid phase diagrams change as a function of both Lennard-Jones diameter ratio (σ_{11}/σ_{22}) and well-depth ratio ($\epsilon_{11}/\epsilon_{22}$).

All of the phase diagrams presented in this section were calculated at a reduced pressure, $P^* = 0.002$, which corresponds to a real pressure of 1 atm using the Lennard-Jones parameters for argon. Due to the large number of mixtures investigated, we did not estimate error bars for the simulation data but we expect the uncertainty to be similar to the uncertainty estimated for the simulation data in Figures A.2-A.4.

To begin our investigation, we examined two limiting cases: sizes are equal ($\sigma_{11}/\sigma_{22} = 1$) while attractions are varied, and attractions are equal ($\epsilon_{11}/\epsilon_{22} = 1$)

while sizes are varied. The ultimate limiting case, a binary mixture with σ_{11}/σ_{22} = 1 and $\epsilon_{11}/\epsilon_{22}$ = 1, is uninteresting; the temperature-composition phase diagram would simply be a horizontal line at $T^* = 0.687$ since the components are indistinguishable. Phase diagrams were calculated for binary mixtures with a diameter ratio σ_{11}/σ_{22} = 1 and well-depth ratios $\epsilon_{11}/\epsilon_{22}$ = 0.65, 0.75, 0.85, 0.90, and 0.95. Figure A.5 shows temperature-composition phase diagrams for these binary mixtures. All of the mixtures form solid solutions with a spindle shape. As $\epsilon_{11}/\epsilon_{22}$ decreases (attractions among species 2 particles become stronger), the melting point temperature of species 2 increases and the degree of phase separation (width of the spindle) increases.

We also calculated phase diagrams for binary mixtures with a well-depth ratio $\epsilon_{11}/\epsilon_{22}$ = 1 and diameter ratios σ_{11}/σ_{22} = 0.85, 0.86, 0.87, 0.88, 0.9, 0.91, 0.93, 0.94, and 0.95. Figure A.6 shows selected temperature-composition phase diagrams for these binary mixtures. An azeotrope phase diagram is obtained for σ_{11}/σ_{22} = 0.95. As σ_{11}/σ_{22} decreases from 0.95 to 0.86, the azeotrope temperature decreases and the degree of phase separation on either side of the azeotrope composition increases. The azeotrope temperature decreases as σ_{11}/σ_{22} decreases because for increasing size differences, the liquid phase becomes more stable than the solid phase.[49] At σ_{11}/σ_{22} = 0.86, the solidus line takes a steep drop in temperature over the composition range x_2 = 0 to 0.05, and as a result, the shape of the azeotrope appears distorted. At σ_{11}/σ_{22} = 0.85 the solidus line is no longer continuous across the entire range of composition and a eutectic phase diagram is obtained. Thus,

we conclude that an azeotrope to eutectic transition occurs between $\sigma_{11}/\sigma_{22} = 0.85$ and 0.86. In comparison, Kranendonk and Frenkel[19] found an azeotrope to eutectic transition for hard-sphere mixtures at $\sigma_{11}/\sigma_{22} = 0.875$.

Next, we attempted to establish boundaries for solid solution, azeotrope, and eutectic phase behavior in $\epsilon_{11}/\epsilon_{22}$ versus σ_{11}/σ_{22} parameter space. We calculated five series of phase diagrams at diameter ratios $\sigma_{11}/\sigma_{22} = 0.85$, 0.875, 0.9, 0.925, and 0.95. Within each σ_{11}/σ_{22} series, the well-depth ratio $\epsilon_{11}/\epsilon_{22}$ was varied from 0.45 to 1.6.

In the first series (not shown), we calculated phase diagrams for binary mixtures with $\sigma_{11}/\sigma_{22} = 0.95$ and $\epsilon_{11}/\epsilon_{22} = 0.55, 0.65, 0.75, 1.0, 1.1, 1.2, 1.3, 1.4,$ and 1.6. We found solid solution phase diagrams for all the mixtures in this series, except at $\epsilon_{11}/\epsilon_{22} = 1.0$ where, as described above, the phase diagram displayed a minimum azeotrope. As $\epsilon_{11}/\epsilon_{22}$ moves away from unity, the width of the spindle increases. A hard-sphere mixture with this diameter ratio has a solid solution phase diagram.

In the second series, we calculated phase diagrams for binary mixtures with $\sigma_{11}/\sigma_{22} = 0.925$ and $\epsilon_{11}/\epsilon_{22} = 0.625, 0.8, 1.0, 1.2, 1.4$ and 1.6. Figure A.7 shows selected temperature-composition phase diagrams for binary mixtures from this series. An azeotrope phase diagram is obtained for $\epsilon_{11}/\epsilon_{22} = 1.0$. As $\epsilon_{11}/\epsilon_{22}$ decreases from 1 to 0.8, the melting point of pure component 2 increases and the azeotrope temperature increases. When the azeotrope temperature is equal to the melting point temperature of pure component 1 ($\epsilon_{11}/\epsilon_{22} = 0.625$), the azeotrope disappears and the phase diagram becomes a solid solution. Conversely, as $\epsilon_{11}/\epsilon_{22}$ increases

from 1 to 1.2, the melting point temperature of pure component 2 decreases and the azeotrope temperature decreases. When the azeotrope is equal to the melting point of pure component 2 ($\epsilon_{11}/\epsilon_{22} > 1.2$), the azeotrope disappears and the phase diagram becomes a solid solution. As $\epsilon_{11}/\epsilon_{22}$ increases from 1.2 to 1.6, the melting point of species 2 decreases and the degree of phase separation increases. A hard-sphere mixture with this diameter ratio has an azeotrope phase diagram.

In the third series, we calculated phase diagrams for binary mixtures with $\sigma_{11}/\sigma_{22} = 0.9$ and $\epsilon_{11}/\epsilon_{22} = 0.45, 0.65, 0.85, 0.9, 1.0, 1.05, 1.1, 1.2, 1.4,$ and 1.6. Figure A.8 shows selected temperature-composition phase diagrams for binary mixtures from this series. An azeotrope phase diagram is obtained for $\epsilon_{11}/\epsilon_{22} = 1.0$. As $\epsilon_{11}/\epsilon_{22}$ decreases from 1 to 0.65, the melting point of pure component 2 increases and the azeotrope temperature increases. As $\epsilon_{11}/\epsilon_{22}$ decreases from 0.65 to 0.45, the azeotrope disappears and the phase diagram becomes a simple peritectic. A simple peritectic phase diagram is characterized by a peritectic temperature at which the liquid composition is either greater than or less than the compositions of the two solid phases with which it coexists. The inset in Figure A.8 shows the region of the peritectic phase diagram between the compositions, $x_2 = 0 - 0.1$. As $\epsilon_{11}/\epsilon_{22}$ increases from 1 to 1.4, the melting point temperature of pure component 2 decreases and the azeotrope temperature decreases. When the azeotrope temperature is equal to the melting point of pure component 2, the azeotrope disappears and the phase diagram becomes a solid solution. As $\epsilon_{11}/\epsilon_{22}$ increases from 1.4 to 1.6, the phase diagram changes from a solid solution to a simple peritectic. A hard-sphere

mixture with this diameter ratio has an azeotrope phase diagram.

In the fourth series, we calculated phase diagrams for binary mixtures with $\sigma_{11}/\sigma_{22} = 0.875$ and $\epsilon_{11}/\epsilon_{22} = 0.625, 0.8, 1, 1.2, 1.4,$ and 1.6. Figure A.9 shows selected temperature-composition phase diagrams for binary mixtures from this series. An azeotrope phase diagram is obtained for $\epsilon_{11}/\epsilon_{22} = 1.0$. As $\epsilon_{11}/\epsilon_{22}$ decreases from 1 to 0.625, the melting point of pure component 2 increases and the azeotrope temperature increases. As $\epsilon_{11}/\epsilon_{22}$ increases from 1 to 1.2, the melting point temperature of pure component 2 decreases and the azeotrope temperature decreases. At $\epsilon_{11}/\epsilon_{22} = 1.2$, the solidus line takes a steep drop in temperature over the composition range $x_2 = 0$ to 0.1 and as a result, the shape of the azeotrope appears distorted. As $\epsilon_{11}/\epsilon_{22}$ increases from 1.2 to 1.6, the phase diagram changes from an azeotrope to a eutectic. A hard-sphere mixture with this diameter ratio shows a transition between azeotrope and eutectic phase behavior.

In the fifth and final series, we calculated phase diagrams for binary mixtures with $\sigma_{11}/\sigma_{22} = 0.85$ and $\epsilon_{11}/\epsilon_{22} = 0.625, 0.8, 0.9, 1.0, 1.1,$ and 1.6. Figure A.10 shows selected temperature-composition phase diagrams for binary mixtures from this series. A eutectic phase diagram is obtained for $\epsilon_{11}/\epsilon_{22} = 1.0$. As $\epsilon_{11}/\epsilon_{22}$ decreases from 1 to 0.625, the melting point of pure component 2 increases and the eutectic temperature increases. As $\epsilon_{11}/\epsilon_{22}$ increases from 1 to 1.6, the melting point temperature of pure component 2 decreases and the eutectic temperature decreases. A hard-sphere mixture with this diameter ratio has a eutectic phase diagram.

In Figure A.11, we show the regions of σ_{11}/σ_{22} vs. $\epsilon_{11}/\epsilon_{22}$ parameter space

where the four types of solid-liquid phase diagrams (solid solutions, azeotropes, eutectics, and simple peritectics) are found for the Lennard-Jones model. Each mixture studied in this paper is given a symbol corresponding to the type of phase diagram calculated via a Gibbs-Duhem integration simulation. We did not calculate phase diagrams for $\sigma_{11}/\sigma_{22} = 1$ and $\epsilon_{11}/\epsilon_{22} > 1$ because due to symmetry the phase diagrams in this region will be identical to the phase diagrams for $\epsilon_{11}/\epsilon_{22} < 1$.

Using these simulation results, boundaries separating the regions of solid solution, azeotrope, and eutectic phase behavior have been estimated. Boundaries for the peritectic region have not been estimated because simple peritectics occurred only a few times. Between $\sigma_{11}/\sigma_{22} = 0.95$ and 1, solid solution type phase diagrams are found everywhere except for a narrow region near $\epsilon_{11}/\epsilon_{22} = 1$ where azeotropes are formed because the melting points of the pure components are nearly equal[25]. As σ_{11}/σ_{22} decreases below 0.95, this azeotropic region broadens and then narrows again as σ_{11}/σ_{22} approaches 0.86, forming a fish-shaped region whose maximum height spans the range $\epsilon_{11}/\epsilon_{22} = 0.6$ to 1.2. The region outside of the azeotrope region switches from solid solution to eutectic as the diameter ratio σ_{11}/σ_{22} decreases below 0.9.

To understand how and why variations in the attractive interactions lead to the boundaries shown in the $\epsilon_{11}/\epsilon_{22}$ versus σ_{11}/σ_{22} parameter space displayed in Figure A.11, it is helpful to consider the schematic phase diagrams for the selected mixtures shown in Figure A.12. The columns correspond to mixtures with constant diameter ratios, $\sigma_{11}/\sigma_{22} = 0.875$, 0.9, and 0.925, and the rows correspond to

mixtures with constant well-depth ratios, $\epsilon_{11}/\epsilon_{22}$ = 0.625, 1.0, and 1.6. The solid-liquid lines are based on our simulation results. The solid-solid equilibrium lines are based on our best guess as to how the upper critical solution temperature of the solid-solid immiscibility dome shifts with variations in well-depth ratio. Underpinning this guess is quasi-chemical theory[50,51], which tells us that the upper critical solution temperature increases as the attractions between like molecules become stronger than the attractions between unlike molecules. This means that the upper critical solution temperature will increase as $\epsilon_{11}/\epsilon_{22}$ increases above unity or decreases below unity. This is shown schematically in Figure A.12.

In the σ_{11}/σ_{22} = 0.925 column, the phase diagram is a solid solution at $\epsilon_{11}/\epsilon_{22}$ = 1.6. As the well-depth ratio $\epsilon_{11}/\epsilon_{22}$ decreases, the melting point temperature of component 2 increases and the solid-liquid lines "pull away" from the solid-solid immiscibility dome. In the σ_{11}/σ_{22} = 0.9 column, the phase diagram is a simple peritectic at $\epsilon_{11}/\epsilon_{22}$ = 1.6. As the well-depth ratio decreases, the solid-liquid lines move away from the underlying solid-solid immiscibility dome (shown in dashed lines) and the phase diagram becomes an azeotrope. In the σ_{11}/σ_{22} = 0.875 column, the phase diagram is a eutectic at $\epsilon_{11}/\epsilon_{22}$ = 1.6. As the well-depth ratio decreases, the solid-liquid lines move away from the solid-solid immiscibility dome and the phase diagram becomes an azeotrope.

In the $\epsilon_{11}/\epsilon_{22}$ = 1.6 row, the phase diagram is a solid solution at σ_{11}/σ_{22} = 0.925. As the diameter ratio decreases, the solid-liquid lines "fall" because the liquid phase can accommodate the larger differences in size easier than the solid phase[49] and the

phase diagram becomes a simple peritectic at $\sigma_{11}/\sigma_{22} = 0.9$ and a eutectic at $\sigma_{11}/\sigma_{22} = 0.875$. In the $\epsilon_{11}/\epsilon_{22} = 1.0$ row, the phase diagram is an azeotrope at $\sigma_{11}/\sigma_{22} = 0.925$. As the diameter ratio decreases and the solid-liquid lines fall, the azeotrope temperature decreases. In the $\epsilon_{11}/\epsilon_{22} = 0.625$ row, the phase diagram is a solid solution at $\sigma_{11}/\sigma_{22} = 0.925$. As the diameter ratio decreases and the solid-liquid lines fall, the phase diagram becomes an azeotrope.

A.4 Summary

The Gibbs-Duhem integration technique was combined with semigrand canonical Monte Carlo simulations to calculate solid-liquid phase diagrams for binary Lennard-Jones mixtures.

We calculated phase diagrams for model mixtures: argon-methane, argon-krypton, and krypton-methane, and compared them to cell theory predictions and experimental data. We found that cell theory qualitatively predicts the shape of phase diagrams for Lennard-Jones mixtures. Comparison of our simulation results for the argon-krypton Lennard-Jones system with hard sphere results at the same diameter ratio ($\sigma_{Ar}/\sigma_{Kr} = 0.93$) indicated that the presence of attractive interactions can change the type of phase diagram observed from azeotrope (hard sphere) to solid solution (Lennard-Jones). This suggested that we further explore the effect of asymmetry in the attractions on the types of phase diagrams observed.

We simulated 56 binary Lennard-Jones mixtures over a range of diameter ratios $\sigma_{11}/\sigma_{22} = 0.85 - 1.0$ and well-depth ratios $\epsilon_{11}/\epsilon_{22} = 0.45 - 1.6$. We found that for

well-depth ratios of unity (equal attractions among species), phase behavior indicative of azeotropes and eutectics is observed for diameter ratios ranging from 0.85 to 1. Much to our surprise, we also found that varying the attractive interactions at a fixed diameter ratio can perturb the type of solid-liquid phase diagram obtained, a trend that previously[6,8] had not been explored in Lennard-Jones mixtures.

There are several areas for further study. First, it would be helpful to know where the solid-solid immiscible dome is located relative to the solid-liquid lines that we have calculated in this paper. Determining solid-solid equilibrium lines would give a more complete picture of how the solid-liquid and solid-solid equilibrium lines merge as we vary parameters and observe transitions between solid solutions, azeotropes, eutectics, and simple peritectics. Second, it would be interesting to repeat this type of investigation for $\sigma_{11}/\sigma_{22} < 0.85$, a region where several ordered solid phases are known to be stable.

A.5 References

[1] M. Matsuoka, Bunri Gijutsu, **7**, 245–249 (1977).

[2] X. Cottin and P. A. Monson, Journal of Chemical Physics, **99**(11), 8914–8921 (1993).

[3] X. Cottin and P. A. Monson, Journal of Chemical Physics, **102**(8), 3354–3360 (1995).

[4] X. Cottin. *A theoretical study of the thermodynamics of solid solutions and solid-liquid phase equilibrium*. PhD thesis, University of Massachusetts, (1996).

[5] X. Cottin, E. P. A. Paras, C. Vega, and P. A. Monson, Fluid Phase Equilibria, **117**, 114–125 (1996).

[6] X. Cottin and P. A. Monson, Journal of Chemical Physics, **105**(22), 10022–10029 (1996).

[7] D. W. Oxtoby, Nature, **347**, 725–730 (1990).

[8] S. W. Rick and A. D. J. Haymet, Journal of Chemical Physics, **90**(2), 1188–1199 (1989).

[9] S. W. Rick and A. D. J. Haymet, Journal of Physical Chemistry, **94**(13), 5212–5220 (1990).

[10] X. C. Zeng and D. W. Oxtoby, Journal of Chemical Physics, **93**(6), 4357–4363 (1990).

[11] A. R. Denton and N. W. Ashcroft, Physical Review A, **42**(12), 7312–7329 (1990).

[12] D. Frenkel and B. Smit, *Understanding Molecular Simulations*. (Academic Press, Inc, San Diego, 1996).

[13] D. Frenkel and A. J. C. Ladd, Journal of Chemical Physics, **81**(7), 3188–3193 (1984).

[14] R. L. Rowley, *Statistical Mechanics for Thermophysical Property Calculations*. (PTR Prentice Hall, Engelwood Cliffs, New Jersey, 1994).

[15] M. D. Eldridge, P. A. Madden, and D. Frenkel, Nature, **365**, 35–37 (1993).

[16] M. D. Eldridge, P. A. Madden, and D. Frenkel, Molecular Physics, **79**(1), 105–120 (1993).

[17] M. D. Eldridge, P. A. Madden, and D. Frenkel, Molecular Physics, **80**(4), 987–995 (1993).

[18] W. G. T. Kranendonk and D. Frenkel, Molecular Physics, **72**(3), 699–713 (1991).

[19] W. G. T. Kranendonk and D. Frenkel, Molecular Physics, **72**(3), 679–697 (1991).

[20] D. A. Kofke, Molecular Simulation, **7**, 285–302 (1991).

[21] D. A. Kofke and E. D. Glandt, Journal of Chemical Physics, **87**(8), 4881–4890 (1987).

[22] D. A. Kofke and E. D. Glandt, Molecular Physics, **64**(6), 1105–1131 (1988).

[23] W. G. Hoover and F. H. Ree, Journal of Chemical Physics, **49**(8), 3609–3613 (1968).

[24] M. J. Vlot, J. C. van Miltenburg, H. A. J. Oonk, and J. P. van der Eerden, Journal of Chemical Physics, **107**(23), 10102–10111 (1997).

[25] J. S. Rowlinson and F. L. Swinton, *Liquids and Liquid Mixtures*. (Butterworth Scientific, London, 1982).

[26] M. J. Vlot and J. P. van der Eerden, Journal of Chemical Physics, **106**(7), 2771–2776 (1997).

[27] K. S. Shing and K. E. Gubbins, Molecular Physics, **46**(5), 1109–1128 (1982).

[28] M. J. Vlot, S. Claassen, H. E. A. Huitema, and J. P. van der Eerden, Molecular Physics, **91**(1), 19–30 (1997).

[29] B. Widom, Journal of Chemical Physics, **39**(11), 2808–2812 (1963).

[30] J. M. Prausnitz, R. N. Lichtenthaler, and E. G. de Azevedo, *Molecular Thermodynamics of Fluid-Phase Equilibria*. (Prentice Hall, Englewood Cliffs, New Jersey, second edition, 1986).

[31] A. Z. Panagiotopoulos, Molecular Physics, **61**(4), 813–826 (1987).

[32] A. Z. Panagiotopoulos, N. Quirke, M. Stapleton, and D. J. Tildesley, Molecular Physics, **63**(4), 527–545 (1988).

[33] A. Z. Panagiotopoulos, Molecular Simulation, **9**(1), 1 (1992).

[34] D. A. Kofke, Molecular Physics, **78**(6), 1331–1336 (1993).

[35] D. A. Kofke, Journal of Chemical Physics, **98**(5), 4149–4162 (1993).

[36] R. Agrawal, M. Mehta, and D. A. Kofke, International Journal of Thermophysics, **15**(6), 1073–1083 (1994).

[37] R. Agrawal and D. A. Kofke, Molecular Physics, **85**(1), 43–59 (1995).

[38] M. H. J. Hagen, E. J. Meijer, G. C. A. M. Mooij, D. Frenkel, and H. N. W. Lekkerkerker, Nature, **365**, 425–426 (1993).

[39] E. J. Meijer and D. Frenkel, Journal of Chemical Physics, **100**(9), 6873–6887 (1994).

[40] F. W. Tavares and S. I. Sandler, AIChE Journal, **43**(1), 218–231 (1997).

[41] D. A. Kofke, in *Monte Carlo Methods in Chemistry* edited by D. M. FergusonJ. I. Siepmann and D. G. Truhlar (Interscience Publishers, New York, 1998).

[42] M. Mehta and D. A. Kofke, Chemical Engineering Science, **49**(16), 2633–2645 (1994).

[43] B. Carnahan, H. A. Luther, and J. O. Wilkes, *Applied Numerical Methods*. (John Wiley & Sons, New York, 1969).

[44] M. P. Allen and D. J. Tildesley, *Computer Simulation of Liquids*. (Clarendon Press, Oxford, 1987).

[45] J.-P. Hansen and L. Verlet, Physical Review, **184**, 151–161 (1969).

[46] P. van't Zelfde, M. H. Omar, H. G. M. le Pair-Schroten, and Z. Dokoupil, Physica, **38**, 241–252 (1968).

[47] H. Veith and E. Schroder, Z. Physik. Chem. A, **179**, 16 (1937).

[48] R. Heastie, Nature, **176**(4485), 747–748 (1955).

[49] W. Hume-Rothery, R. E. Smallman, and C. W. Haworth, *The Structure of Metals and Alloys*. (The Metals and Metallurgy Trust, London, 1969).

[50] A. Prince, *Alloy Phase Equilibria*. (Elsevier, Amsterdam, 1966).

[51] P. Gordon, *Principles of Phase Diagrams in Materials Systems*. (Krieger, Malabar, Florida, 1983).

Table A.1: Diameter ratios for binary hard-sphere mixtures that exhibit solid-liquid phase diagrams of the solid solution, azeotrope, and eutectic types. All of the solid phases are substitutionally disordered solid solutions except for eutectics with complete immiscibility, in which case the solid phases contain only one component.

Phase diagram	Cell theory[4]	Thermodynamic integration[19]	Semigrand Monte Carlo[20]
Solid solution	0.95, 0.97	0.95	0.95
Solid solution/ azeotrope boundary		0.9425	
Azeotrope	0.9, 0.92, 0.93, 0.94	0.9, 0.92	0.9, 0.93
Azeotrope/ eutectic boundary		0.875	
Eutectic with partial immiscibility	0.85	0.85	
Eutectic with complete immiscibility	0.62, 0.73, 0.8		

Table A.2: Range of diameter ratios at which binary hard-sphere mixtures form stable substitutionally ordered solid solutions of the types AB_{13}, AB_2, and AB.

Solid structure	Cell theory[4]	Einstein crystal[16,17]
AB_{13}	0.54 - 0.61	0.474 - 0.626
AB_2	0.42 - 0.59	0.4 - 0.62
AB	0.2 - 0.42	-

Table A.3: Lennard-Jones potential parameters for argon, krypton, and methane.

	ϵ_{ii}/k(K)	$\sigma_{ii}(A)$
Ar	141.2	3.336
Kr	191.4	3.575
CH4	159.7	3.706

A.6 Figures

Page

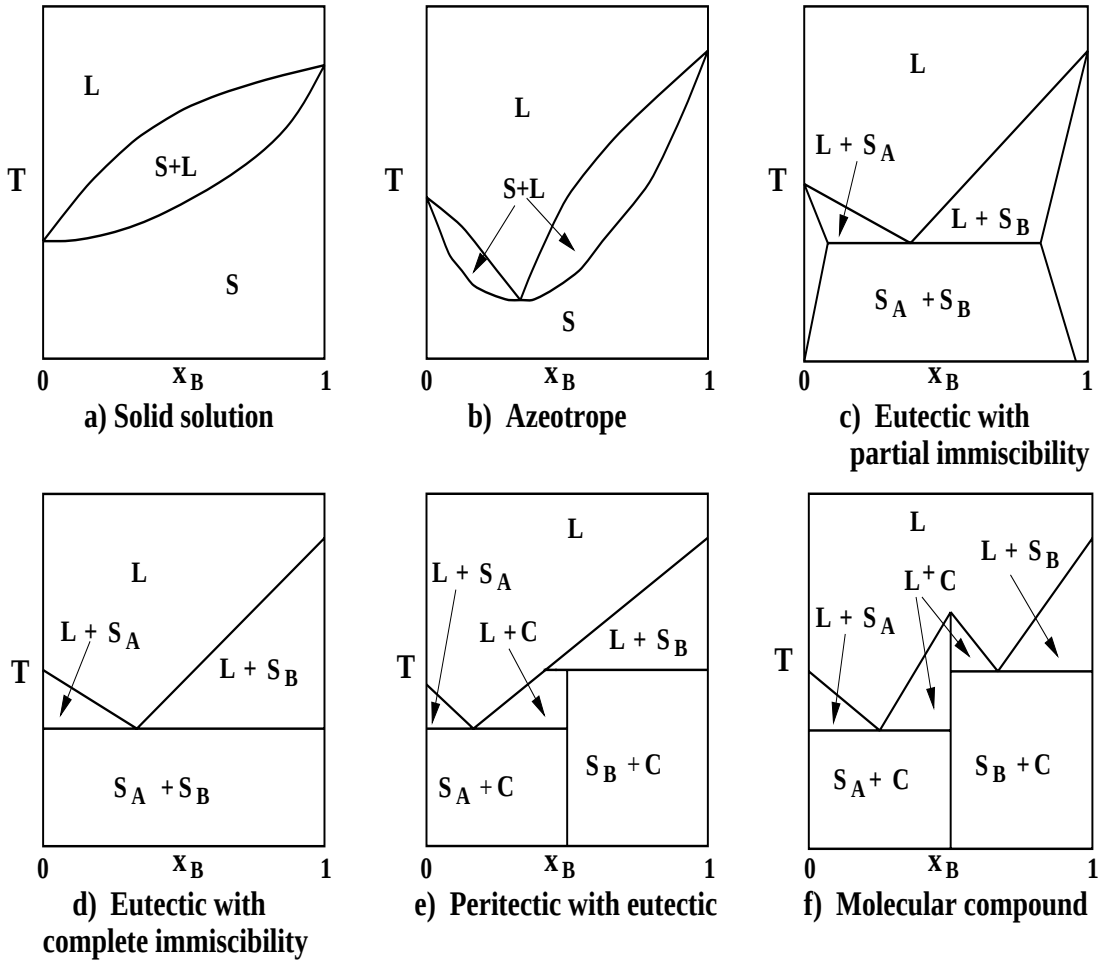

Figure A.1: The six types of solid-liquid phase diagrams identified by Matsuoka[1]: a) solid solution, b) azeotrope, c) eutectic with partial immiscibility, d) eutectic with complete immiscibility, e) peritectic with eutectic, and f) molecular compound. The phase diagrams are shown in the $T - x_B$ plane where x_B is the mole fraction of component B. The symbols are: L = liquid mixture of A and B, S = solid solution of A and B, S_A = solid solution rich in A, S_B = solid solution rich in B, C = ordered solid with fixed stoichiometric ratio A_mB_n.

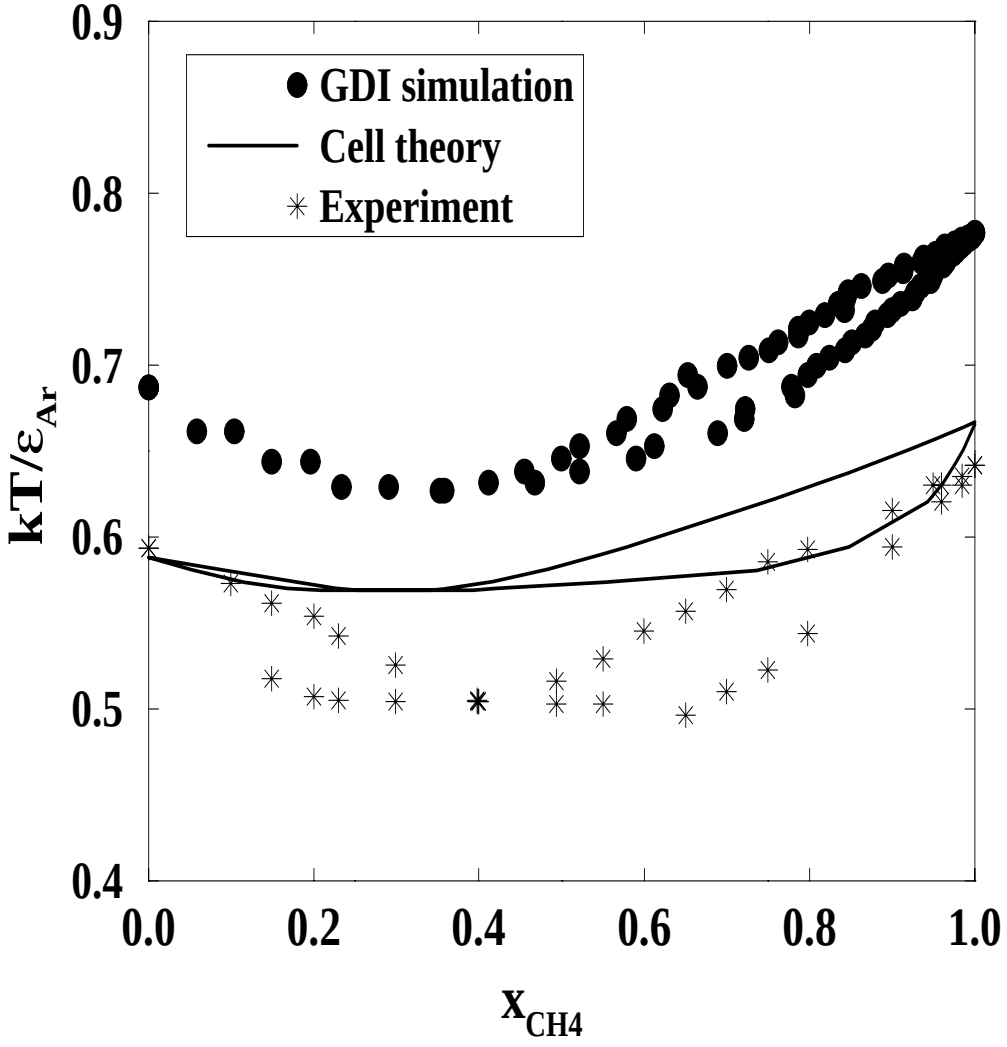

Figure A.2: Temperature vs. composition solid-liquid phase diagram for the argon-methane system at 1 atm. The filled circles represent Gibbs-Duhem integration (GDI) simulations, the solid line corresponds to the Cottin and Monson's cell theory[6] prediction, and the stars correspond to van't Zelfde *et al.*[46] experiments.

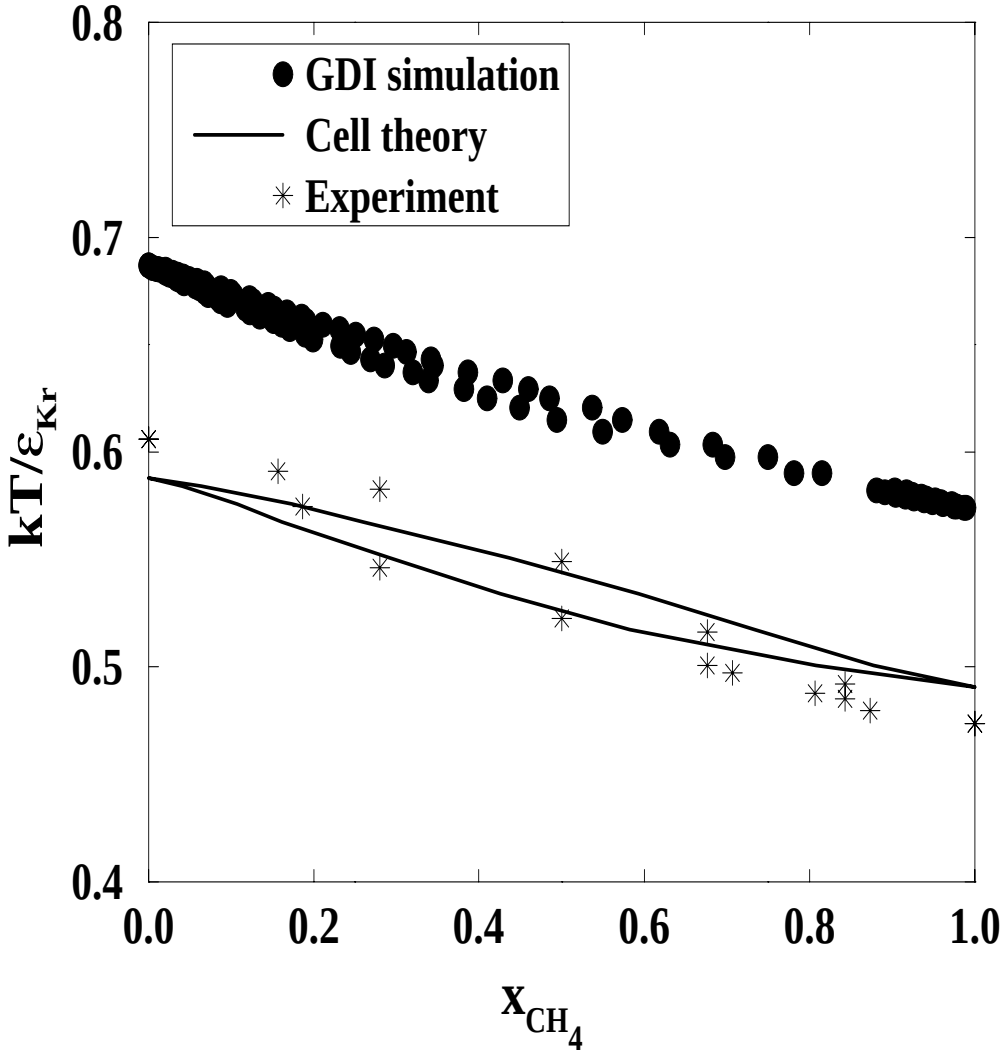

Figure A.3: Temperature vs. composition solid-liquid phase diagram for the krypton-methane system at 1 atm. The filled circles represent Gibbs-Duhem integration (GDI) simulations, the solid line corresponds to the Cottin and Monson's cell theory[6] prediction, and the stars correspond to Veith and Schroder's[47] experiments.

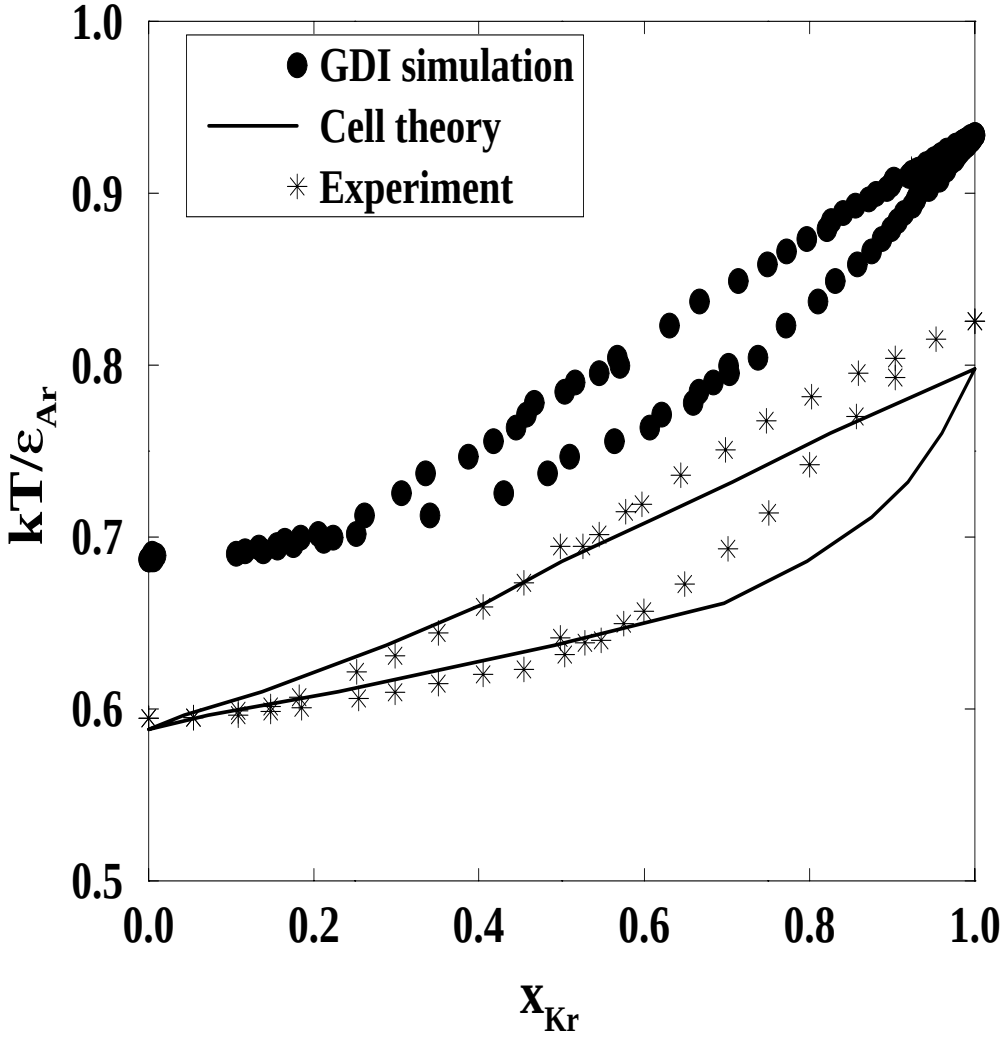

Figure A.4: Temperature vs. composition solid-liquid phase diagram for the argon-krypton system at 1 atm. The filled circles represent Gibbs-Duhem integration (GDI) simulations, the solid line corresponds to the Cottin and Monson's cell theory[6] prediction, and the stars correspond to Heastie's[48] experiments.

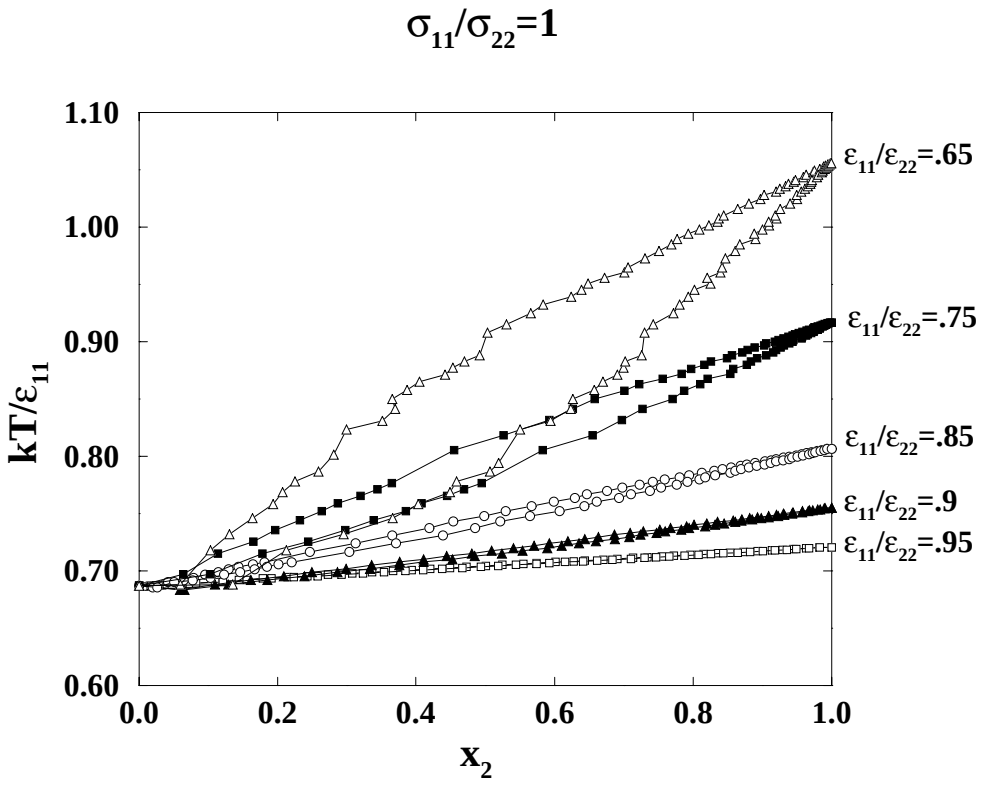

Figure A.5: Temperature vs. composition solid-liquid phase diagrams for Lennard-Jones binary mixtures at diameter ratio, σ_{11}/σ_{22} = 1.0 and well-depth ratios $\epsilon_{11}/\epsilon_{22}$ = 0.65, 0.75, 0.85, 0.90, and 0.95. Lines are drawn through the simulation points for clarity.

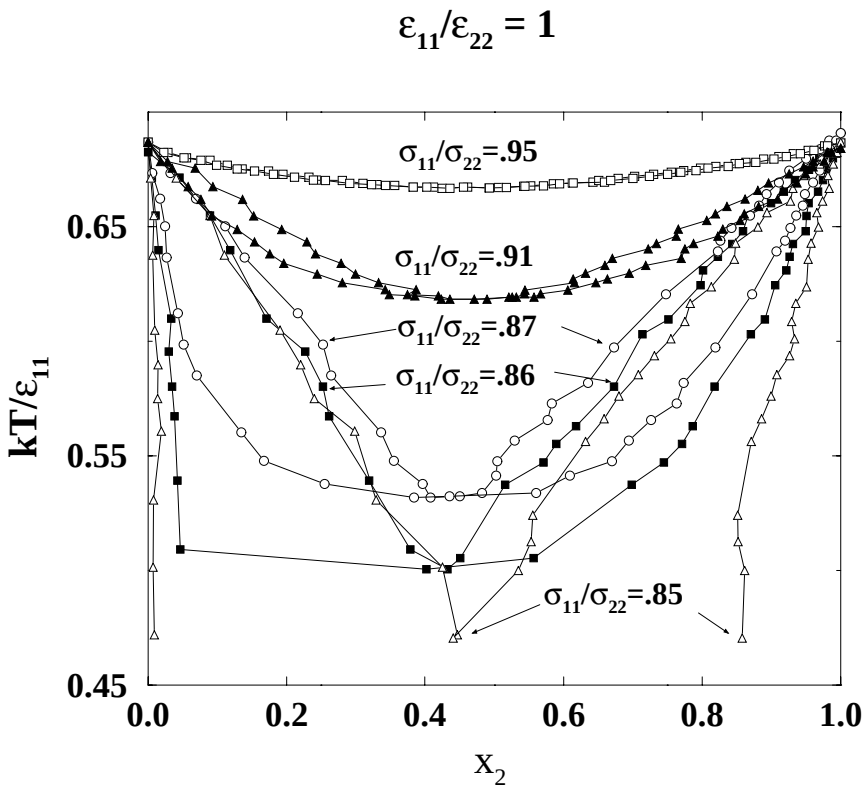

Figure A.6: Temperature vs. composition solid-liquid phase diagrams for Lennard-Jones binary mixtures at well-depth ratio $\epsilon_{11}/\epsilon_{22}$ = 1.0 and diameter ratios, σ_{11}/σ_{22} = 0.85, 0.86, 0.87, 0.91, and 0.95.

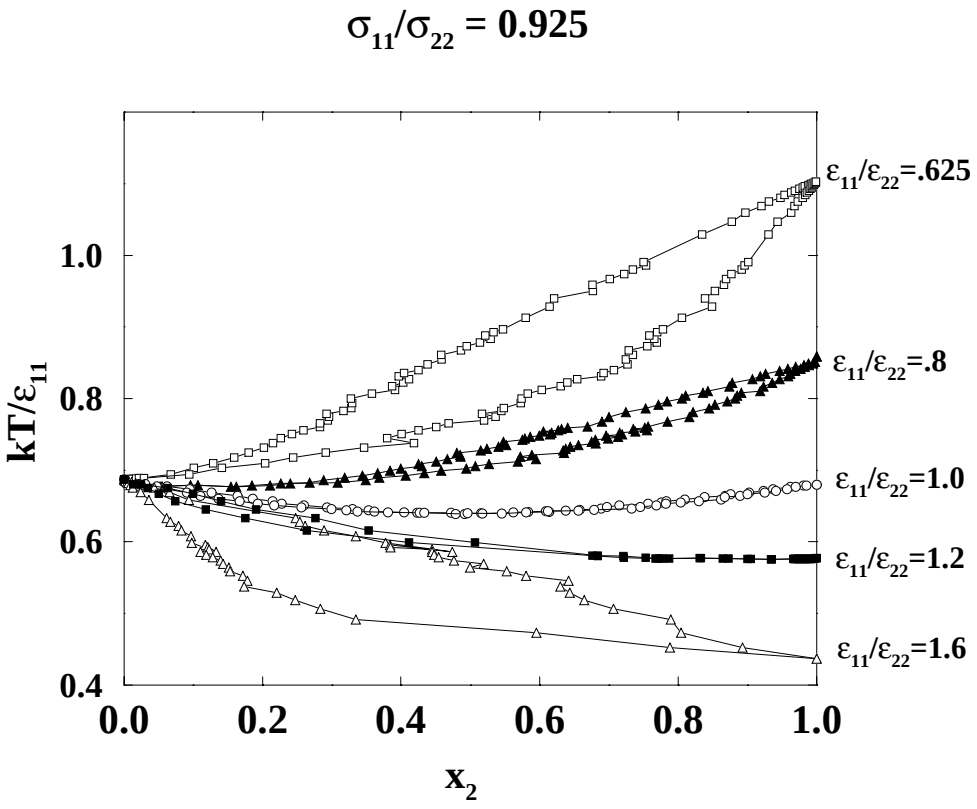

Figure A.7: Temperature vs. composition solid-liquid phase diagrams for Lennard-Jones binary mixtures at diameter ratio, σ_{11}/σ_{22} = 0.925 and well-depth ratios $\epsilon_{11}/\epsilon_{22}$ = 0.625, 0.8, 1.0, 1.2, and 1.6.

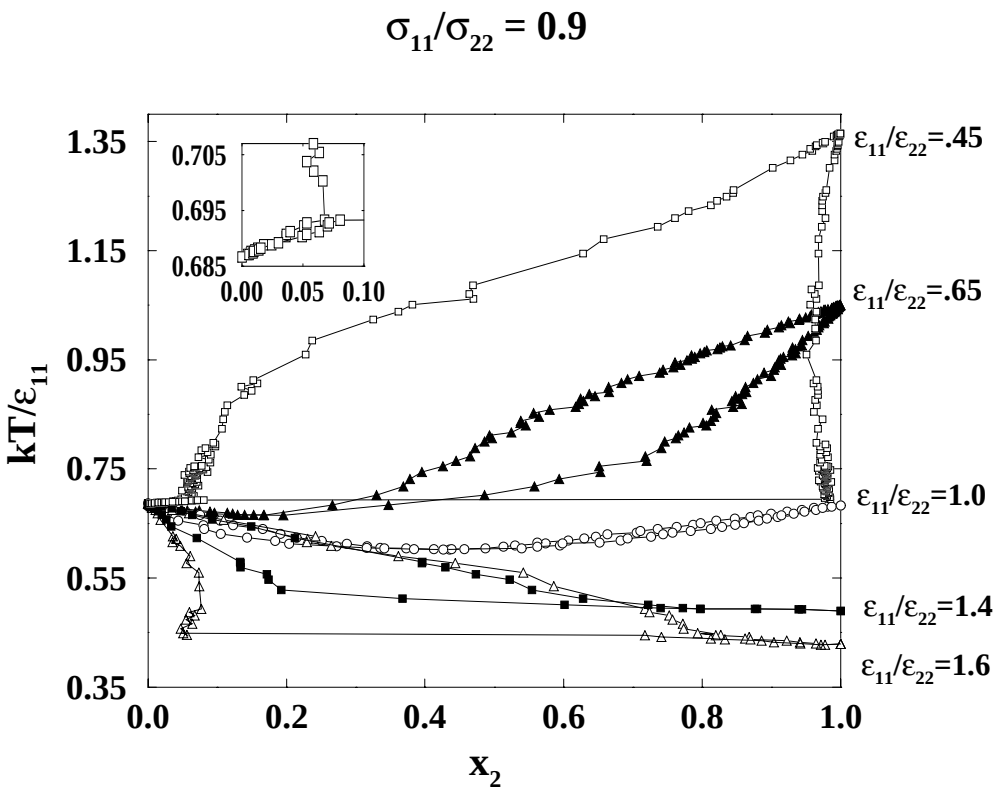

Figure A.8: Temperature vs. composition solid-liquid phase diagrams for Lennard-Jones binary mixtures at diameter ratio, $\sigma_{11}/\sigma_{22} = 0.9$ and well-depth ratios $\epsilon_{11}/\epsilon_{22}$ = 0.45, 0.65, 1.0, 1.4, and 1.6. The inset shows the region around the peritectic temperature at $x_2 = 0 - 0.1$ for $\epsilon_{11}/\epsilon_{22} = 0.45$.

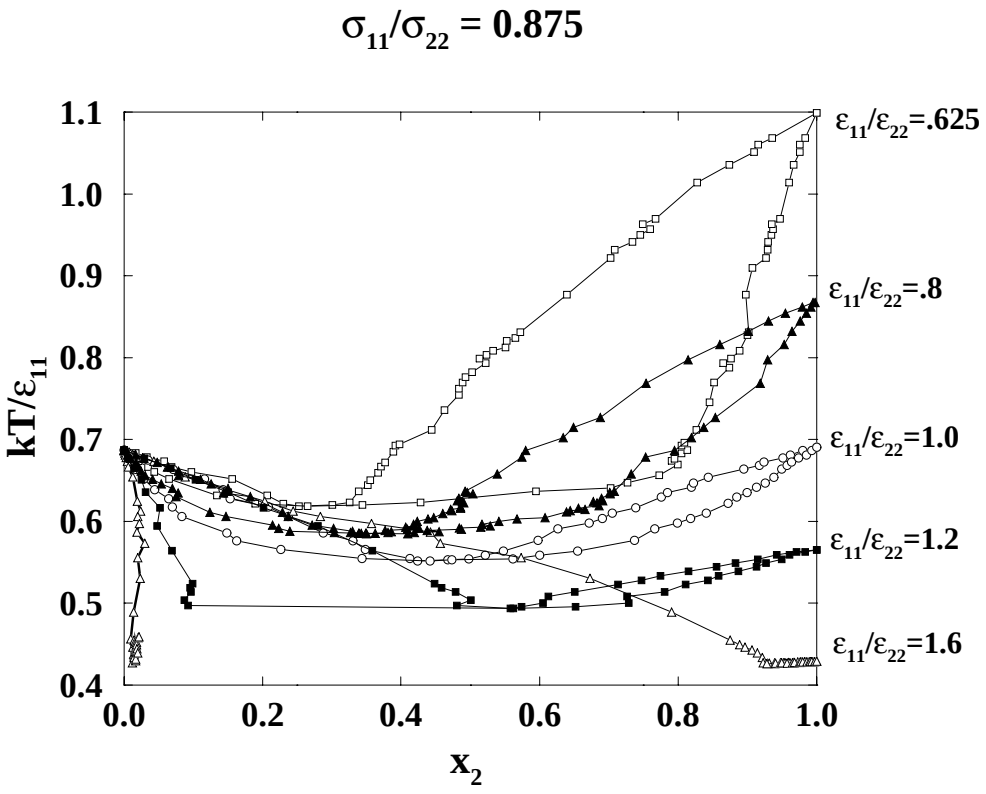

Figure A.9: Temperature vs. composition solid-liquid phase diagrams for Lennard-Jones binary mixtures at diameter ratio, σ_{11}/σ_{22} = 0.875 and well-depth ratios $\epsilon_{11}/\epsilon_{22}$ = 0.625, 0.8, 1.0, 1.2, and 1.6.

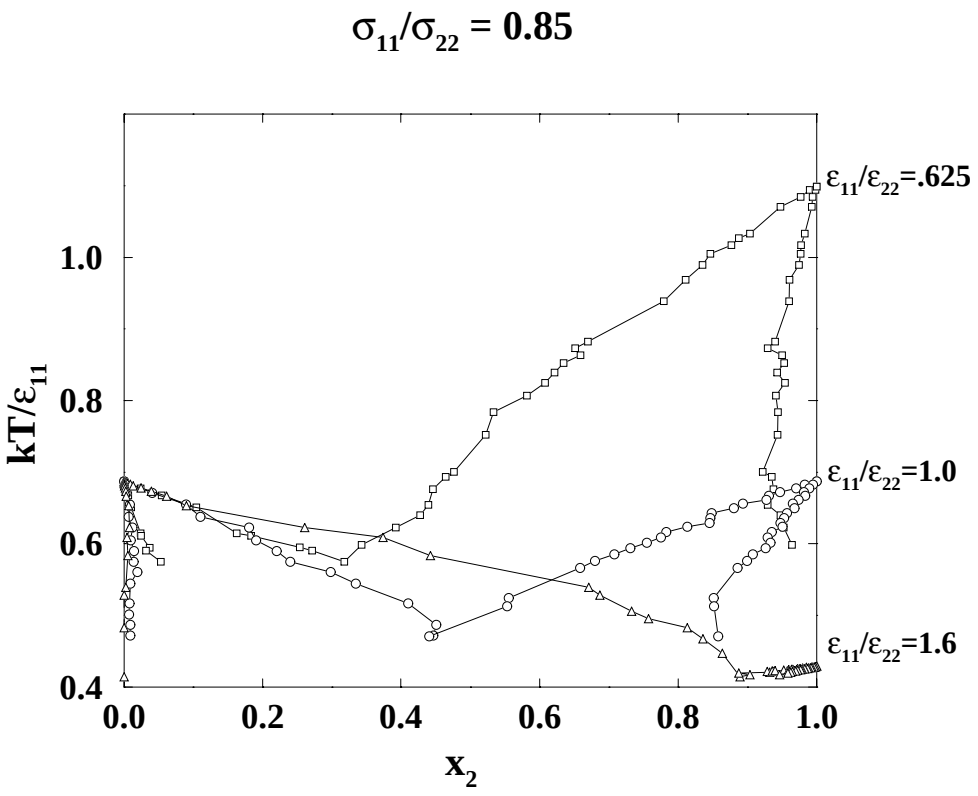

Figure A.10: Temperature vs. composition solid-liquid phase diagrams for Lennard-Jones binary mixtures at diameter ratio, $\sigma_{11}/\sigma_{22} = 0.85$ and well-depth ratios $\epsilon_{11}/\epsilon_{22}$ = 0.625, 1.0, and 1.6.

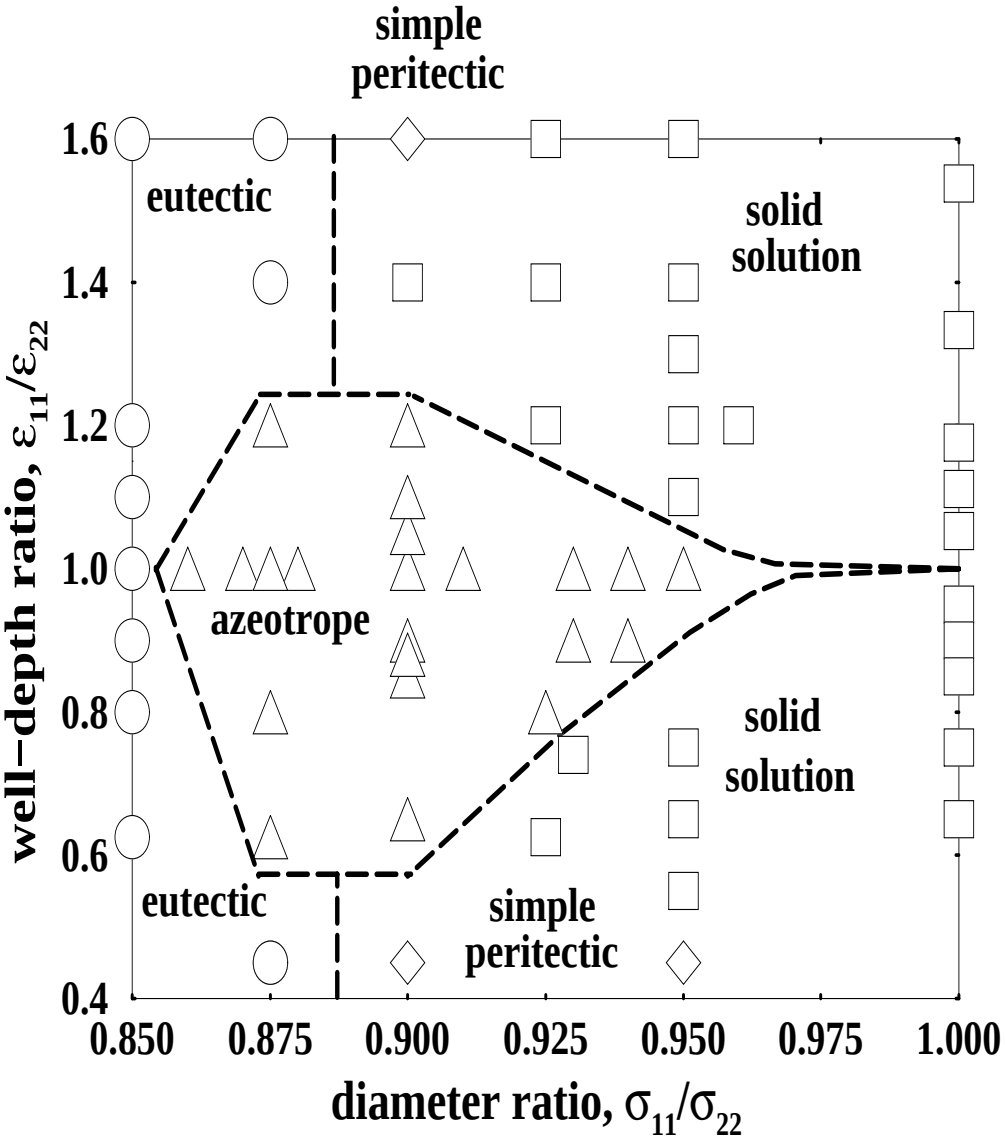

Figure A.11: Boundaries (dashed lines) separating regions of solid solutions (squares), azeotropes (triangles), eutectics (circles), and simple peritectics (diamonds) in Lennard-Jones parameter space, $0.45 \leq \epsilon_{11}/\epsilon_{22} \leq 1.6$ and $0.85 \leq \sigma_{11}/\sigma_{22} \leq 1.0$.

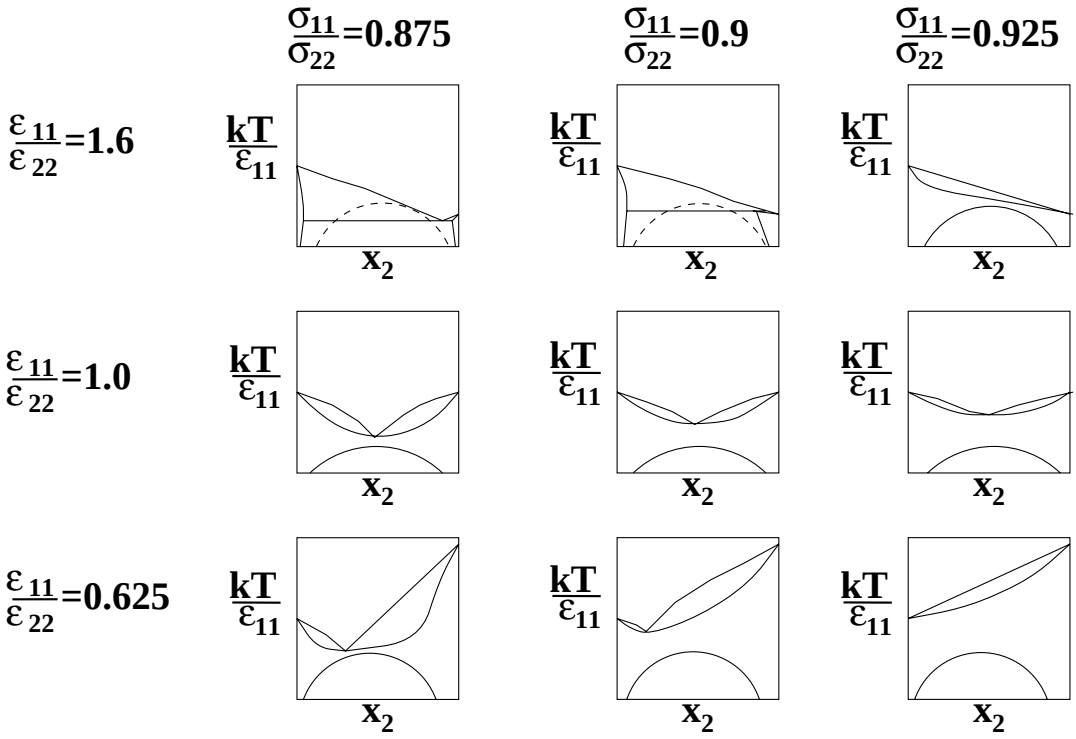

Figure A.12: Schematic phase diagrams showing how variations in diameter ratio and well-depth ratio lead changes in the phase diagram. The columns correspond to mixtures with constant diameter ratios, σ_{11}/σ_{22} = 0.875, 0.9, and 0.925, and the rows correspond to mixtures with constant well-depth ratios, $\epsilon_{11}/\epsilon_{22}$ = 0.625, 1.0, and 1.6. The solid-liquid lines were calculated in this work. The solid-solid lines have not been calculated for these mixtures, but we have guessed how the upper critical solution temperature shifts with variation in well-depth ratio using quasi-chemical theory.

APPENDIX B

CODE LISTING

In this Appendix we provide the source code for the computer program that was used to calculate temperature versus composition phase diagrams via Gibbs-Duhem integration. The program is written in FORTRAN 77.

GDI_SLE_P This program uses Gibbs-Duhem integration to calculate a constant pressure solid-liquid phase diagram (temperature versus composition) for a binary Lennard-Jones mixture.

mcnpt_a This subroutine performs the semigrand canonical (NPTξ_2) Monte Carlo simulation for the liquid phase. It was adapted from the NPT code given by Allen and Tildesley, 1987.

mcnpt_b This subroutine performs the semigrand canonical (NPTξ_2) Monte Carlo simulation for the fcc crystalline solid phase. It was adapted from the NPT code given by Allen and Tildesley, 1987.

B.1 Code listing for Gibbs-Duhem integration program

```
program GDI_SLE_P

c   August 30, 1999

c  This program uses Gibbs-Duhem integration to calculate a constant
c  pressure solid-liquid phase diagram (T-x) for a binary LJ mixture.

#include "mainhead3.f"

c*** Modifications

c All real*8 variables are placed in the same block. (8/12/97)

c Put in general dependent/independent integration variables.(11/13/97)

c Added higher order predict/correct eqns (Mehta&Kokfe 94).(11/13/97)

c Added output file for integrand information (run.int). (6/5/98)

c Generate output files suitable for use in xmgr application. (7/1/98)

c Changed the # of decimal places in the format statements. (7/8/98)

c Added INTEGRAND function for calculation F(simcount). (5/19/99)

c Standard deviation, block averages. (8/25/99)

c Variable stepsize corrector (Escobedo and DePablo, JCP 1997).
c (8/26/99)


c***  Comments   ***

c The index "simcount" is the simulation counter. (simcount=0 for
c initial condition, simcount=1 for first simulation, etc.)

c The index "update" is the dependent variable update counter.
c It counts how many times the simulation averages are used to
c update the dependent variable.



c*****************************  Start code ******************************
c*************************************************************************
c234567890123456789012345678901234567890123456789012345678901234567890123456789012

C*** Put this statement at the beginning of what you want to time
DELTA = DTIME(TARRAY)

c Read in data file (initial condition).
call rd_data(input,npoints,hen2_a,hen2_b,
     :                           hh,maxlimit,minlimit)

c Read in particle configuration files for each phase.

cnfile='liquid.xyz'

call readcn_a(cnfile)
```

```
cnfile='solid.xyz'

call readcn_b(cnfile)


c These files are for recording the program output.

open(unit=17,file=run//'_liq',status='unknown')
open(unit=18,file=run//'_sol',status='unknown')
write(17,*)'#   x_l          T        +/- x_a'
write(18,*)'#   x_v          T        +/- x_b'


open(unit=7,file=run//'.out',status='unknown')
WRITE(7, *)'****************************************'
WRITE(7, *)
WRITE(7, *) 'Lennard-Jones parameters'
WRITE(7, *) '*** phase_a ***'


do i=1,NCOMPA
do j=1,NCOMPA
WRITE(7,110) i,j,eps_a(i,j),i,j,sigmas_a(i,j)
enddo
enddo

WRITE(7,*) '*** phase_b ***'

do i=1,NCOMPB
do j=1,NCOMPB
WRITE(7,110) i,j,eps_b(i,j),i,j,sigmas_b(i,j)
enddo
enddo


c Write system parameters

WRITE(7, *)
WRITE(7,90) 'integration stepsize ',hh(0)
WRITE(7,10) 'number of data points',npoints
WRITE(7, *)
WRITE(7,*) "Number of attempted interchanges, NSWAP",NSWAP
WRITE(7,*)
WRITE(7,*) "Henry's law constant, liquid",hen2_a
WRITE(7,*) "Henry's law constant, solid",hen2_b
WRITE(7,*)
WRITE(7,*) "maxlimit, minlimit"
WRITE(7,*) maxlimit, minlimit
WRITE(7,*)
        WRITE(7,20)
WRITE(7,30)
WRITE(7,40) pstep_a,pstep_b
WRITE(7,50) cstep_a,cstep_b
WRITE(7,60) ncorr,ncorr
WRITE(7,70) fstep_a,fstep_b
WRITE(7, *)
WRITE(7,80) NOPA,NOPB
WRITE(7, *)
WRITE(7, *) 'Initial conditions'
WRITE(7, *)
WRITE(7,90) 'temperature          ',TEMP
WRITE(7,90) 'pressure             ',P
```

```
WRITE(7, *)
WRITE(7,25)
WRITE(7,30)
WRITE(7,95) 'volume               ',vol_a,vol_b
WRITE(7,95) 'enthalpy             ',h_a,h_b


 10     FORMAT(' ',A22,'   ',I7)
 20 FORMAT(' ','parameter             phase_a      phase_b')
 25 FORMAT(' ','variable              phase_a      phase_b')
 30     FORMAT(' ','---------             -------      -------')
 40     FORMAT(' ','predictor cycles      ',I7,'      ',I7)
 50     FORMAT(' ','corrector cycles      ',I7,'      ',I7)
 60     FORMAT(' ','corrector iterations  ',I7,'      ',I7)
 70     FORMAT(' ','production cycles     ',I7,'      ',I7)
 80     FORMAT(' ','total particles       ',I7,'      ',I7)
 90     FORMAT(' ',A22,'   ',F8.6)
 95 FORMAT(' ',A22,                   F10.4,'      ',F10.4)
 100    FORMAT(' ','epsilon = ',F4.2,'    sigma = ',F4.2)
 110    FORMAT(' ','eps(',I1,I1,') = ',F5.3,'   sig(',I1,I1,') = ',F5.3)
 120    FORMAT(5X,'xi2',4X,'TEMP',3X,'x_a(2)',2X,'x_b(2)',2X,'vol_a',2x,
     :         'vol_b',3X,'h_a',5X,'h_b')

WRITE(7, *) 'fugacity fraction, species 2'
WRITE(7,90) 'xi2                  ',fug_frac
c Calculate mole fraction of each species
c     phase_a

x_a(1)=REAL(N_a(1))/REAL(NOPA)
        x_a(2)=REAL(N_a(2))/REAL(NOPA)
WRITE(7,90) 'x_a(2)               ',x_a(2)
c     FILL IN 'species' ARRAY
c     The species array holds a 1 or 2 to identify each particle.

      do  i=1,NCOMPA
         do  index=N_a(i-1)+1,N_a(i-1)+N_a(i)
            species_a(index) = i
   enddo
      enddo
c     PRELIM CALCULATIONS ON LJ PARAMETERS
      do  i=1,NCOMPA
         do  j=1,NCOMPA
            eps_a4(i,j) = 4.0*eps_a(i,j)
            sig2_a(i,j) = sigmas_a(i,j)*sigmas_a(i,j)
            sig3_a(i,j) = sig2_a(i,j) *sigmas_a(i,j)
         enddo
      enddo

c     phase_b

x_b(1)=REAL(N_b(1))/REAL(NOPB)
x_b(2)=REAL(N_b(2))/REAL(NOPB)
WRITE(7,90) 'x_b(2)               ',x_b(2)
write(7,*)
write(7,*)
write(7,120)
c     FILL IN 'species' ARRAY
c     The species array holds a 1 or 2 to identify each particle.

      do  i=1,NCOMPB
         do  index=N_b(i-1)+1,N_b(i-1)+N_b(i)
            species_b(index) = i
   enddo
```

```
      enddo
c     PRELIM CALCULATIONS ON LJ PARAMETERS
      do  i=1,NCOMPB
         do  j=1,NCOMPB
            eps_b4(i,j) = 4.0*eps_b(i,j)
            sig2_b(i,j) = sigmas_b(i,j)*sigmas_b(i,j)
            sig3_b(i,j) = sig2_b(i,j) *sigmas_b(i,j)
         enddo
      enddo


c***** Use initial condition to estimate next coexistence point ********
c**********************************************************************
c234567890123456789012345678901234567890123456789012345678901234567890123456789012

c Calculate G-D integrand using initial condition information.

c two, binary component phases

  depend(0) =  1/TEMP
  indpnd(0) = fug_frac

#ifdef HENRY
        F(0) = (1/hen2_a-1/hen2_b)/(h_a-h_b)
#else
        F(0) = (x_a(2)-x_b(2))/(indpnd(0)*(1-indpnd(0))*(h_a-h_b))
#endif

c  Write integrand to file (for keeping track of step size).
open(unit=27,file=run//'.int',status='unknown')
                write(27,*) indpnd(0),F(0),hh(0)

open(unit=37,file=run//'corr.log',status='unknown')
write(37,*) simcount
write(37,*) '# update     BETA'

c Increment independent variable.
indpnd(1) = indpnd(0)+ hh(0)


c Use trapezoid predictor to estimate new dependent variable.
depend(1) = depend(0) + hh(0)*F(0)


c************************** Start the main loop************************
c**********************************************************************

c One complete loop gives one phase point.
c Input npoints from main input file.


do 150 simcount = 1,npoints


c *** PREDICTOR CYCLES ***
SFLAG = 1
c two, binary component phases

fug_frac = indpnd(simcount)
TEMP = 1/depend(simcount)

c      zero accumulators and averages
update=0
```

```
c -----phase_a
 tens = INT(simcount/10)
         ones = MOD(simcount,10)
infile=input//'_a.pred'
  outfile=run//CHAR(tens+48)//CHAR(ones+48)//'_a.pred'
c        bfile=run//CHAR(tens+48)//CHAR(ones+48)//'_ap.block'
xfilea=run//CHAR(tens+48)//CHAR(ones+48)//'_a.comp'
pfilea=run//CHAR(tens+48)//CHAR(ones+48)//'_a.pres'
dfilea=run//CHAR(tens+48)//CHAR(ones+48)//'_a.rho'
c outfile=run//'_a.pred'
        bfile=run//'_ap.block'
call mcnpt_a(pstep_a,infile,outfile,bfile,update,SFLAG)

c -----phase_b
infile=input//'_b.pred'
  outfile=run//CHAR(tens+48)//CHAR(ones+48)//'_b.pred'
c bfile=run//CHAR(tens+48)//CHAR(ones+48)//'_bp.block'
xfileb=run//CHAR(tens+48)//CHAR(ones+48)//'_b.comp'
pfileb=run//CHAR(tens+48)//CHAR(ones+48)//'_b.pres'
dfileb=run//CHAR(tens+48)//CHAR(ones+48)//'_b.rho'
c outfile=run//'_b.pred'
  bfile=run//'_bp.block'
call mcnpt_b(pstep_b,infile,outfile,bfile,update,SFLAG)

c  Calculate integrand of G-D equation based on
c   averages from predictor cycles.

c two, binary component phases

      F(simcount) = (avx_a(2)-avx_b(2))/(indpnd(simcount)*
     :                  (1-indpnd(simcount))*(h_a-h_b))

c *** CORRECTOR CYCLES ***
SFLAG = 2

c      zero accumulators and averages
c       when update .le. 1

c      apply corrector
        if(simcount.eq.1) then
c        trapezoid corrector

         depend(simcount) = depend(simcount-1) +
     :                 hh(simcount-1)/2. *( F(simcount) + F(simcount-1))

else
c        2nd order corrector, 1st variant

r=hh(simcount-2)/hh(simcount-1)
A=0.0
B=1.0
Cminus1=-(1./r)
Cnaught= (1./r) +4. +3.*r
Cplus1= 2.+3.*r
D = hh(simcount-1)/6./(1.+r)

depend(simcount) = A*depend(simcount-2)+B*depend(simcount-1)
     :   + D*(Cminus1*F(simcount-2)+Cnaught*F(simcount-1)
     :           +Cplus1*F(simcount))

c        elseif(simcount.eq.2) then
c        second corrector
c     depend(simcount) = depend(simcount-2) +
```

```
c    :                hh/3. *(F(simcount)+4.*F(simcount-1)+
c    :                F(simcount-2))

c       else
c        third corrector and beyond
c
c        depend(simcount) = depend(simcount-1) +
c    :                hh/24. *(9.*F(simcount)+19.*F(simcount-1)+
c    :                -5.*F(simcount-2)+F(simcount-3))

        endif


c      Loop over corrector iterations
c      Input ncorr from main input file.

write(37,*) update, depend(simcount)
           do 140 update = 1,ncorr

c two, binary component phases

TEMP = 1/depend(simcount)
c -----phase_a
infile=input//'_a.corr'
c outfile=run//CHAR(tens+48)//CHAR(ones+48)//'_a.corr'
c bfile=run//CHAR(tens+48)//CHAR(ones+48)//'_ac.block'
outfile=run//'_a.corr'
bfile=run//'_ac.block'
call mcnpt_a(cstep_a,infile,outfile,bfile,update,SFLAG)
c -----phase_b
infile=input//'_b.corr'
c outfile=run//CHAR(tens+48)//CHAR(ones+48)//'_b.corr'
c bfile=run//CHAR(tens+48)//CHAR(ones+48)//'_bc.block'
outfile=run//'_b.corr'
bfile=run//'_bc.block'
call mcnpt_b(cstep_b,infile,outfile,bfile,update,SFLAG)

c  Calculate integrand of G-D equation based on
c   averages from corrector cycles (including all previous iterations).

c two, binary component phases

       F(simcount) = (avx_a(2)-avx_b(2))/(indpnd(simcount)*
     :                  (1-indpnd(simcount))*(h_a-h_b))

c      apply corrector
        if(simcount.eq.1) then
c        trapezoid corrector

         depend(simcount) = depend(simcount-1) +
     :                hh(simcount-1)/2. *( F(simcount) + F(simcount-1))

else
c        2nd order corrector, 1st variant

depend(simcount) = A*depend(simcount-2)+B*depend(simcount-1)
     :   + D*(Cminus1*F(simcount-2)+Cnaught*F(simcount-1)
     :          +Cplus1*F(simcount))

endif

write(37,*) update, depend(simcount)
```

```
 140         continue

write(37,*)
write(37,*)
write(37,*) simcount
write(37,*) '# update     BETA'

c *** PRODUCTION CYCLES ***
SFLAG = 3
c two, binary component phases

TEMP = 1/depend(simcount)

c      zero accumulators and averages
update=0
c -----phase_a
infile=input//'_a.prod'
c outfile=run//CHAR(tens+48)//CHAR(ones+48)//'_a.prod'
  bfile=run//CHAR(tens+48)//CHAR(ones+48)//'_af.block'
outfile=run//'_a.prod'
c bfile=run//'_af.block'
call mcnpt_a(fstep_a,infile,outfile,bfile,update,SFLAG)
c -----phase_b
infile=input//'_b.prod'
c outfile=run//CHAR(tens+48)//CHAR(ones+48)//'_b.prod'
  bfile=run//CHAR(tens+48)//CHAR(ones+48)//'_bf.block'
outfile=run//'_b.prod'
c bfile=run//'_bf.block'
call mcnpt_b(fstep_b,infile,outfile,bfile,update,SFLAG)

c Write output to file.

 WRITE(7,'(I2,4(1X,F7.4),2(1X,F10.1),2(1X,F7.3))')
     :                    simcount,indpnd(simcount),TEMP,avx_a(2),
     :                     avx_b(2),
     :                    vol_a,vol_b,h_a,h_b

c output file for XMGR
         write(17,*) avx_a(2),TEMP,stdx2_a
         write(18,*) avx_b(2),TEMP,stdx2_b

c  Write positions to file to record frames.

c posfilea=run//CHAR(tens+48)//CHAR(ones+48)//'_apos.xyz'
c posfileb=run//CHAR(tens+48)//CHAR(ones+48)//'_bpos.xyz'
c         call writpos_a(posfilea)
c call writpos_b(posfileb)
posfilea=run//CHAR(tens+48)//CHAR(ones+48)//'_a.xyz'
posfileb=run//CHAR(tens+48)//CHAR(ones+48)//'_b.xyz'
        call writcn_a(posfilea)
call writcn_b(posfileb)


c  Calculate integrand of G-D equation based on
c   averages from production cycles.

c two, binary component phases

    F(simcount) = (avx_a(2)-avx_b(2))/(indpnd(simcount)*
     :                   (1-indpnd(simcount))*(h_a-h_b))

c  Write integrand to file (for keeping track of step size)
         write(27,*) indpnd(simcount),F(simcount),hh(simcount-1)
```

```
product = hh(simcount-1)*F(simcount)

if( ABS(product) .gt. maxlimit) then
hh(simcount) = hh(simcount-1)*maxlimit/ABS(product)
        elseif( ABS(product) .lt. minlimit) then
  hh(simcount) = hh(simcount-1)*minlimit/ABS(product)
else
      hh(simcount) = hh(simcount-1)
endif

c    Step to next indpnd
       indpnd(simcount+1) =indpnd(simcount) + hh(simcount)

if ( indpnd(simcount+1) .ge. 1.0) then
 write(7,*) "Fugacity fraction is greater than 1",indpnd(simcount+1)
C*** Put this statement again at the end of what you want to time
DELTA = DTIME(TARRAY)

C*** DELTA is in seconds - change to hours (or minutes) - calc collision rate
EXTIME = DELTA/3600
WRITE(7,'('' EXECUTION TIME (CPU HOURS)  '',F15.4)') EXTIME
 STOP
endif

if ( indpnd(simcount+1) .le. 0.0) then
 write(7,*) "Fugacity fraction is less than 0",indpnd(simcount+1)
C*** Put this statement again at the end of what you want to time
DELTA = DTIME(TARRAY)

C*** DELTA is in seconds - change to hours (or minutes) - calc collision rate
EXTIME = DELTA/3600
WRITE(7,'('' EXECUTION TIME (CPU HOURS)  '',F15.4)') EXTIME
 STOP
endif


c-------predicted depend = depend(i+1)
    depend(simcount+1) = depend(simcount)
     :         + hh(simcount)*F(simcount)


 150     continue

C*** Put this statement again at the end of what you want to time
DELTA = DTIME(TARRAY)

C*** DELTA is in seconds - change to hours (or minutes) - calc collision rate
EXTIME = DELTA/3600
WRITE(7,'('' EXECUTION TIME (CPU HOURS)  '',F15.4)') EXTIME

10002   FORMAT(//1X,'  i   ..depend(i)..  .. T(i) .. ..vol_a..',
     :             '.. vol_b.. ..h_a.. ..h_b..  ..x_a(2). ..')

stop

end

#include "rd_data.f"
#include"readcn_a.f"
#include"writcn_a.f"
#include"mcnpt_a.f"
#include"sumup_a.f"
```

```
#include"energy_a.f"
#include"writpos_a.f"
#include"energi_a.f"
#include"readcn_b.f"
#include"writcn_b.f"
#include"mcnpt_b.f"
#include"sumup_b.f"
#include"energy_b.f"
#include"writpos_b.f"
#include"energi_b.f"
#include"integrand.f"
```

```
      subroutine mcnpt_a(nstep,infile,outfile,bfile,j,SFLAG)

c August 30, 1999
c MRH  corrections are in lower case.
c A&T original code in upper case.

c Changed random number generator to  drandm (9/10/96).
c Make choice of mc move with a certain probability. (7/24/97)
c Select atom to displace at random.                  (7/24/97)
c Select volume change with probability 5/(NOP+5).    (11/26/97)
c Corrected mistake in longrange correction virial term
c    term3w, (4/14/99)
c Remove CPP directives. (5/24/99)
c Sample properties at the end of each step. (5/24/99)

c234567890123456789012345678901234567890123456789012345678901234567890123456789012
C     ************************************************************************
C     ** THIS FORTRAN CODE IS INTENDED TO ILLUSTRATE POINTS MADE IN    **
C     ** THE TEXT. TO OUR KNOWLEDGE IT WORKS CORRECTLY. HOWEVER IT IS  **
C     ** THE RESPONSIBILITY OF THE USER TO TEST IT, IF IT IS USED IN A **
C     ** RESEARCH APPLICATION.                                         **
C     ************************************************************************

C     ************************************************************************
C     ** FICHE F.12                                                    **
C     ** MONTE CARLO SIMULATION IN THE CONSTANT-NPT ENSEMBLE.          **
C     ************************************************************************

C     ************************************************************************
C     ** MONTE CARLO SIMULATION IN THE CONSTANT-NPT ENSEMBLE.          **
C     **                                                               **
C     ** THIS PROGRAM TAKES A CONFIGURATION OF LENNARD JONES ATOMS     **
C     ** AND PERFORMS A MONTE CARLO SIMULATION AT CONSTANT NPT. THE    **
C     ** BOX IS IN UNITS OF SIGMA THE LENNARD JONES DIAMETER.          **
C     ** THERE ARE NO LOOKUP TABLES INCLUDED.                          **
C     **                                                               **
C     ** REFERENCE:                                                    **
C     **                                                               **
C     ** MCDONALD, CHEM. PHYS. LETT. 3, 241, 1969.                     **
C     **                                                               **
C     ** PRINCIPAL VARIABLES:                                          **
C     **                                                               **
C     ** INTEGER N                          NUMBER OF MOLECULES        **
C     ** REAL    RX(N),RY(N),RZ(N)          POSITIONS                  **
C     ** REAL    VOL                        VOLUME                     **
C     ** REAL    BOX                        BOX LENGTH                 **
C     ** REAL    DENS                       REDUCED DENSITY            **
C     ** REAL    TEMP                       REDUCED TEMPERATURE        **
C     ** REAL    SIGMA                      REDUCED LJ DIAMETER        **
C     ** REAL    DRMAX                      MAXIMUM DISPLACEMENT       **
C     ** REAL    V                          THE POTENTIAL ENERGY       **
C     ** REAL    W                          THE VIRIAL                 **
C     ** REAL    PRESUR                     REQUIRED PRESSURE          **
C     ** REAL    DBOXMX                     MAX CHANGE IN BOX          **
c     ** real       h                                 enthalpy
C     **                                                               **
C     ** USAGE:                                                        **
C     **                                                               **
C     ** CONDUCTS MONTE CARLO SIMULATION AT CONSTANT PRESSURE FOR A    **
C     ** SPECIFIED NUMBER OF CYCLES FROM A GIVEN INITIAL CONFIGURATION.**
```

```
C     **                                                              **
C     ** UNITS:                                                       **
C     **                                                              **
C     ** THIS PROGRAM USES THE USUAL REDUCED LJ UNITS. IN PARTICULAR   **
C     ** THE BOX LENGTH IS IN UNITS OF SIGMA.                         **
C     **                                                              **
C     ** ROUTINES REFERENCED:                                         **
C     **                                                              **
C     ** SUBROUTINE ENERGY ( RXI, RYI, RZI, I, RCUT, BOX, V12, V6,    **
C     **    :                W12, W6 )                                **
C     **    CALCULATES THE ENERGY AND VIRIAL FOR ATOM I IN THE FLUID  **
C     ** SUBROUTINE READCN ( CNFILE, BOX )                            **
C     **    READS IN CONFIGURATION AND BOX VARIABLES                  **
C     ** SUBROUTINE SUMUP ( RCUT, RMIN, OVRLAP, BOX, V12, V6, W12, W6 )**
C     **    CALCULATES POTENTIAL AND VIRIAL FOR A CONFIGURATION       **
C     ** SUBROUTINE WRITCN ( CNFILE, BOX )                            **
C     **    WRITES OUT CONFIGURATION AND BOX VARIABLES                **
C     ****************************************************************


      REAL        RX(NOPA), RY(NOPA), RZ(NOPA)

      INTEGER     STEP, NSTEP, IPRINT, IBLOCK, IRATIO, IRATB, I
      INTEGER     ILOOP,rnd_num
      integer     dcount,vcount
      integer     ACM,ACATMA,ACBOXA
      double precision ACV, ACP, ACD, NORM
      double precision ach,ACVSQ, ACPSQ, ACDSQ
      REAL        AVV, AVP, AVD
      REAL        FLV, FLP, FLD
      REAL        DENS, TEMP, RCUT, RMIN, PRESUR, VOL, PRES, VN
      REAL        BOXINV, BOXNEW, RATBOX, RAT12, RAT6, DVOL, DPV
      REAL        DELTHB, DRMAX, DBOXMX, BETA, DUMMY, RATIO
      REAL        RRBOX, BRATIO, DELTVB, RCUTN
      REAL        RXIOLD, RYIOLD, RZIOLD, RXINEW, RYINEW, RZINEW
      real        RXINEW2, RYINEW2, RZINEW2
      REAL        V12OLD, V6OLD, V12NEW, V6NEW, VS
      REAL        W12OLD, W6OLD, W12NEW, W6NEW, WS, PS
      REAL        DELV12, DELV6, DELW12, DELW6, DELTV
      REAL        V6, W6, V12, W12, VLRC, VLRCN, WLRC, WLRCN
      REAL        SR3, SR9, VLRC6, WLRC6, VLRC12, WLRC12, PI
      real           h,CPX_a
      real        avVOL,BOX,avh
      integer j,ii,jj
      INTEGER IMOVE,NMOVE
      CHARACTER   TITLE*80, CNFILE*30
      character   infile*30, outfile*30, bfile*30
      character   xfile*30,pfile*30,dfile*30
      LOGICAL     OVRLAP

      real NOBS
      real SQV,SQP,SQD,SQH,SQX2
      real SQV_SUM,SQP_SUM,SQD_SUM,SQH_SUM,SQX2_SUM
      real AVV_SUM,AVP_SUM,AVD_SUM,avh_SUM,avx2_SUM,avVOL_SUM
      real STDV,STDP,STDD,STDH,STDX2

c     semigd variables

      integer species(NOPA)
      integer N_a(NCOMPA),Ntry(NCOMPA)
      real    sigmas_a(NCOMPA,NCOMPA),eps_a(NCOMPA,NCOMPA)
      real    sig2_a(NCOMPA,NCOMPA),sig3_a(NCOMPA,NCOMPA)
```

```
        real    eps_a4(NCOMPA,NCOMPA)
        real  RCUT3
        real term3u
        real term3w
        real term9u
        real term9w
        integer ACSWAP,scount,tryiden
        double precision acx(NCOMPA)
        real x_a(NCOMPA),avx(NCOMPA)
        real  sratio,m,xi2
        real xnew(NCOMPA)
        integer NSWAP, SFLAG

c       end of semigd variables


        PARAMETER ( PI = 3.1415927 )

        COMMON / config_a / RX, RY, RZ
        common / syspar / TEMP,PRESUR

        common / block13 / avVOL,BOX,avh

         common / bina1 / sigmas_a,eps_a
         common / bina2 / sig2_a,sig3_a,eps_a4
         common / block2 / species,N_a
         common / block9 / x_a,avx,STDX2
         common / fugacity / xi2
         common / swap / NSWAP
         common / block10 / xfile,pfile,dfile


external time,drandm,srand,rand
        real*8 drandm
        integer*4 iseed
        integer time


c For random number generator
c        iseed = 4
        iseed = time()
d = drandm(iseed)
        call srand(iseed)



C    **********************************************************************

C    ** BASIC SIMULATION PARAMETERS **

         open ( unit = 8, file = infile, status = 'old')

c       RUN TITLE
        READ (8,'(/A7)') TITLE
c       INTERVAL BETWEEN PRINTS IN CYCLES
        READ (8,'(/I7)') IPRINT
c       INTERVAL FOR BLOCK AVERAGES
         READ (8,'(/I7)') IBLOCK
c       INTERVAL FOR UPDATE OF MAXIMUM DISPLACEMENT OF ATOMS IN CYCLES
        READ (8,'(/I7)') IRATIO
c       INTERVAL FOR UPDATE OF MAXIMUM DISPLACEMENT OF THE BOX IN CYCLES
        READ (8,'(/I7)') IRATB
```

```
         close ( unit = 8)

c         if (SFLAG .eq. 1) then
c         open ( unit = 54, file = xfile, status = 'unknown')
c         open ( unit = 64, file = pfile, status = 'unknown')
c         open ( unit = 74, file = dfile, status = 'unknown')
c         endif

c         open ( unit = 4, file = outfile, status = 'unknown')
c         open ( unit = 44, file = bfile, status = 'unknown')


c          write(44,*) 'Block Averages'
c          write(44,10005)

c        write(4,'('' Update = '',I3)') j
c        write(4,*)

c       WRITE(4,'(1H1,'' **** PROGRAM MCSEMI ****              '')')
c       WRITE(4,'(//''MONTE CARLO IN A CONSTANT-N,P,T,xi2 ENSEMBLE  '')')

c        WRITE(4,'(  '' LENNARD-JONES ATOMS                      '')')


c        WRITE(4,'(//1X,A)') TITLE
c        WRITE(4,'('' NUMBER OF CYCLES                = '',I9)') NSTEP
c        WRITE(4,'('' PRINT INTERVAL                  = '',I9)') IPRINT
c        WRITE(4,'('' RATIO UPDATE INTERVAL FOR ATOMS = '',I9)') IRATIO
c        WRITE(4,'('' RATIO UPDATE INTERVAL FOR BOX   = '',I9)') IRATB
c        WRITE(4,'('' DESIRED PRES.                   = '',F10.5)')PRESUR
c        WRITE(4,'('' DESIRED TEMP                    = '',F10.5)')TEMP
c        WRITE(4,'('' Box length                      = '',F10.5)')BOX

c        WRITE(4,'('' Fugacity fraction (species 2)   = '',F10.5)')xi2
c        WRITE(4,'('' Mole fraction (species 2)       = '',F10.5)')x_a(2)


C    ** SET DEPENDENT PARAMETERS **

        VOL    = BOX*BOX*BOX
        BOXINV = 1.0 / BOX
        DENS   = NOPA / VOL
        RCUT   = 0.5*box
c        RCUT=2.5*sigma

        IF ( RCUT .GT. ( 0.5 * BOX ) ) STOP 'CUT-OFF TOO LARGE'

         DBOXMX = BOX / 15.0
c         DRMAX  = 0.5
c        DBOXMX = BOX / 40.0
c drmax = box/1.5


c drmax = box/1.5
        DRMAX  = 0.15


        RMIN   = 0.70
        BETA   = 1.0 / TEMP

c        WRITE(4,'('' INITIAL DENSITY                 = '',F10.5)') DENS
```

```
C     ** CALCULATE LONG-RANGE CORRECTIONS FOR LJ POTENTIAL.     **
C     ** 6 IS FOR ATTRACTIVE CONTRIBUTIONS 12 IS FOR REPULSIVE **

      term3u      = 0.0
      term9u      = 0.0
      term3w      = 0.0
      term9w      = 0.0
      VLRC12      = 0.0
      VLRC6       = 0.0
      WLRC12      = 0.0
      WLRC6       = 0.0

      RCUT3       = RCUT*RCUT*RCUT

      do  ii=1,NCOMPA
         do  jj=1,NCOMPA
            SR3 = sig3_a(ii,jj)/RCUT3
            SR9 = SR3*SR3*SR3

            term9u =  eps_a(ii,jj)* sig3_a(ii,jj)* SR9/9.
            term3u =  eps_a(ii,jj)* sig3_a(ii,jj)*(-SR3)/3.
            term9w = -eps_a(ii,jj)* sig3_a(ii,jj)* SR9*4./3.
            term3w = -eps_a(ii,jj)* sig3_a(ii,jj)*(-2.)*SR3

            VLRC12 = term9u * x_a(ii)* x_a(jj) + VLRC12
            VLRC6  = term3u * x_a(ii)* x_a(jj) + VLRC6
            WLRC12 = term9w * x_a(ii)* x_a(jj) + WLRC12
            WLRC6  = term3w * x_a(ii)* x_a(jj) + WLRC6

         enddo
      enddo

      VLRC12      =  2.* PI* DENS* NOPA* VLRC12
      VLRC6       =  2.* PI* DENS* NOPA* VLRC6
      WLRC12      = -2.* PI* DENS* NOPA* WLRC12/3.
      WLRC6       = -2.* PI* DENS* NOPA* WLRC6/3.

      VLRC   =    VLRC12 + VLRC6
      WLRC   =    WLRC12 + WLRC6

C     ** ZERO ACCUMULATORS **

          ratio = 0.0
          sratio = 0.0
          bratio = 0.0

         if (j.le.1) then
         ACM    = 0
         ACATMA = 0
         ACBOXA = 0

         ACV = 0.0
         ACP = 0.0
         ACD = 0.0
         ach = 0.0

         ACVSQ = 0.0
         ACPSQ = 0.0
         ACDSQ = 0.0

         FLV = 0.0
         FLP = 0.0
```

```
         FLD = 0.0

         dcount = 0
         vcount = 0

         ACSWAP = 0

         do  k=1,NCOMPA
          acx(k) = 0.0
         enddo

         scount = 0

         end if

         AVV_SUM = 0.0
         AVP_SUM = 0.0
         AVD_SUM = 0.0
         avh_SUM = 0.0
         avx2_SUM = 0.0
         avVOL_SUM = 0.0

         SQV_SUM = 0.0
         SQP_SUM = 0.0
         SQD_SUM = 0.0
         SQH_SUM = 0.0
         SQX2_SUM = 0.0


C     ** CALCULATE INITIAL ENERGY AND VIRIAL **

         call sumup_a ( RCUT, RMIN, OVRLAP, V12, V6, W12, W6 )

c         IF ( OVRLAP ) STOP 'OVERLAP IN INITIAL CONFIGURATION'

C     ** CALCULATE THE INITIAL ENERGY AND VIRIAL **

         VS = ( V12 + V6 + VLRC ) /  NOPA
         WS = ( W12 + W6 + WLRC ) / NOPA
         PS = DENS * TEMP + ( W12 + W6 + WLRC ) / VOL

C     ** ADD LONG RANGE CORRECTIONS **
C     ** INTO THE ENERGY AND VIRIAL **

         V12 = V12 + VLRC12
         V6  = V6  + VLRC6
         W12 = W12 + WLRC12
         W6  = W6  + WLRC6

c         WRITE(4,'('' INITIAL V/N                    = '',F10.6)') VS
c         WRITE(4,'('' INITIAL W/N                    = '',F10.6)') WS
c         WRITE(4,'('' INITIAL P                      = '',F10.6)') PS
c         WRITE(4,'(//'' **** START OF MARKOV CHAIN ****'')')


c         WRITE(4,10003)


c         NMOVE = 2*NOPA+1
         NMOVE = NOPA + NSWAP + 1
c         write(*,*) NMOVE
c         i=1
c         if (i.eq.1) STOP 'end of test'
```

```
C     ***********************************************************************
C     ** MAIN LOOP STARTS                                                  **
C     ***********************************************************************

        DO 100 STEP = 1, NSTEP

           DO 50 IMOVE = 1,NMOVE

            rnd_num=INT( drandm(0)*NMOVE ) + 1
c            if(j.eq.1) then
c             write(99,*) rnd_num,IMOVE
c            endif
            if(rnd_num.le.NOPA) then
c           ** trial particle displacement **

              dcount = dcount + 1

c           ** select atom to move **
c               I = rnd_num
               I =INT( drandm(0)*(NOPA-1) ) + 1

              RXIOLD = RX(I)
              RYIOLD = RY(I)
              RZIOLD = RZ(I)

C          ** CALCULATE V FOR AN ATOM IN OLD STATE **


              call energy_a ( RXIOLD, RYIOLD, RZIOLD, I, RCUT,
     :                        V12OLD, V6OLD, W12OLD, W6OLD )


C          ** MOVE ATOM I **
c          leave new positions in uncorrected form

 90           RXINEW = RXIOLD + ( 2.0 * drandm(0) - 1.0 ) * DRMAX
              RYINEW = RYIOLD + ( 2.0 * drandm(0) - 1.0 ) * DRMAX
              RZINEW = RZIOLD + ( 2.0 * drandm(0) - 1.0 ) * DRMAX

c          correct positions to stay in central box
              RXINEW = RXINEW - ANINT ( RXINEW * BOXINV ) * BOX
              RYINEW = RYINEW - ANINT ( RYINEW * BOXINV ) * BOX
              RZINEW = RZINEW - ANINT ( RZINEW * BOXINV ) * BOX


C          ** CALCULATE V FOR ATOM IN NEW STATE **


              call energy_a ( RXINEW, RYINEW, RZINEW, I, RCUT,
     :                        V12NEW, V6NEW, W12NEW, W6NEW )


c if(ovrlap) goto 90

C          ** CHECK FOR ACCEPTANCE **

              DELV12 = V12NEW - V12OLD
              DELV6  = V6NEW  - V6OLD
              DELW12 = W12NEW - W12OLD
              DELW6  = W6NEW  - W6OLD
              DELTV  = DELV12 + DELV6
              DELTVB = BETA * DELTV
```

```
               IF ( DELTVB .LT. 75.0 ) THEN

                  IF ( DELTV .LE. 0.0 ) THEN

                     V12    = V12 + DELV12
                     V6     = V6  + DELV6
                     W12    = W12 + DELW12
                     W6     = W6  + DELW6
                     RX(I)  = RXINEW
                     RY(I)  = RYINEW
                     RZ(I)  = RZINEW
                     ACATMA = ACATMA + 1

                  ELSEIF ( EXP ( - DELTVB ) .GT. drandm (0) ) THEN

                     V12    = V12 + DELV12
                     V6     = V6  + DELV6
                     W12    = W12 + DELW12
                     W6     = W6  + DELW6
                     RX(I)  = RXINEW
                     RY(I)  = RYINEW
                     RZ(I)  = RZINEW
                     ACATMA = ACATMA + 1

                   ENDIF

               ENDIF

C       ** ENDS ATOMS DISPLACEMENT STEP   **


c           elseif(rnd_num.gt.NOPA .and. rnd_num.le.2*NOPA) then
           elseif(rnd_num.gt.NOPA .and. rnd_num.le.(NOPA+NSWAP)) then
c       ** attempt particle identity change **

               scount = scount + 1
c       ** select particle **
c               I = rnd_num - NOPA
                I =INT( drandm(0)*(NOPA-1) ) + 1

               RXIOLD = RX(I)
               RYIOLD = RY(I)
               RZIOLD = RZ(I)

C          ** CALCULATE V FOR A particle IN OLD STATE **

               call energy_a ( RXIOLD, RYIOLD, RZIOLD, I, RCUT,
     :                         V12OLD, V6OLD, W12OLD, W6OLD )

c          ** switch identity of particle **

               if(species(I).eq.1) then
                  tryiden = 2
                  m       = 1.0
                  Ntry(2) = N_a(2) + 1
                  Ntry(1) = N_a(1) - 1
                  xnew(2) = REAL(Ntry(2))/NOPA
                  xnew(1) = REAL(Ntry(1)) /NOPA
               else
                  tryiden = 1
                  m       =-1.0
                  Ntry(1) = N_a(1) + 1
```

```
                  Ntry(2) = N_a(2) - 1
                  xnew(1) = REAL(Ntry(1))/NOPA
                  xnew(2) = REAL(Ntry(2)) /NOPA

               endif

C           ** CALCULATE V FOR A particle IN NEW STATE **

               call energi_a ( RXIOLD, RYIOLD, RZIOLD, I, RCUT,
     :                         V12NEW, V6NEW, W12NEW, W6NEW,tryiden )

c           ** Calculate difference in long ranges corrections **
               term3u      = 0.0
               term9u      = 0.0
               term3w      = 0.0
               term9w      = 0.0
               VLRC12      = 0.0
               VLRC6       = 0.0
               WLRC12      = 0.0
               WLRC6       = 0.0

               RCUT3       = RCUT*RCUT*RCUT

               do  ii=1,NCOMPA
                  do  jj=1,NCOMPA
                     SR3 = sig3_a(ii,jj)/RCUT3
                     SR9 = SR3*SR3*SR3

                     term9u =  eps_a(ii,jj)* sig3_a(ii,jj)* SR9/9.
                     term3u =  eps_a(ii,jj)* sig3_a(ii,jj)*(-SR3)/3.
                     term9w = -eps_a(ii,jj)* sig3_a(ii,jj)* SR9*4./3.
                     term3w = -eps_a(ii,jj)* sig3_a(ii,jj)*(-2.)*SR3

            VLRC12 = term9u *(xnew(ii)*xnew(jj)-x_a(ii)*x_a(jj)) + VLRC12
            VLRC6  = term3u *(xnew(ii)*xnew(jj)-x_a(ii)*x_a(jj)) + VLRC6
            WLRC12 = term9w *(xnew(ii)*xnew(jj)-x_a(ii)*x_a(jj)) + WLRC12
            WLRC6  = term3w *(xnew(ii)*xnew(jj)-x_a(ii)*x_a(jj)) + WLRC6

                  enddo
               enddo

c            ** difference between LRC in old and new state **
               VLRC12      =  2.* PI* DENS* NOPA* VLRC12
               VLRC6       =  2.* PI* DENS* NOPA* VLRC6
               WLRC12      = -2.* PI* DENS* NOPA* WLRC12/3.
               WLRC6       = -2.* PI* DENS* NOPA* WLRC6/3.

C           ** CHECK FOR ACCEPTANCE **

               DELV12 = V12NEW - V12OLD + VLRC12
               DELV6  = V6NEW  - V6OLD + VLRC6
               DELW12 = W12NEW - W12OLD + WLRC12
               DELW6  = W6NEW  - W6OLD + WLRC6
               DELTV  = DELV12 + DELV6
               DELTVB = BETA * DELTV - m*ALOG(xi2/(1.-xi2))

               IF ( DELTVB .LT. 75.0 ) THEN

                  IF ( DELTVB .LE. 0.0 ) THEN

                     V12    = V12 + DELV12
                     V6     = V6  + DELV6
                     W12    = W12 + DELW12
```

```
                  W6     = W6  + DELW6
                  species(I) = tryiden
                  x_a(1) = xnew(1)
                  x_a(2) = xnew(2)
                  N_a(1) = Ntry(1)
                  N_a(2) = Ntry(2)
                  ACSWAP  = ACSWAP + 1

               ELSEIF ( EXP ( - DELTVB ) .GT. drandm(0) ) THEN

                  V12    = V12 + DELV12
                  V6     = V6  + DELV6
                  W12    = W12 + DELW12
                  W6     = W6  + DELW6
                  species(I) = tryiden
                  x_a(1) = xnew(1)
                  x_a(2) = xnew(2)
                  N_a(1) = Ntry(1)
                  N_a(2) = Ntry(2)
                  ACSWAP  = ACSWAP + 1

               ENDIF

            ENDIF

C       ** ENDS particle identity change STEP   **

            else
C              ** ATTEMPT A BOX MOVE **

               vcount = vcount + 1

            BOXNEW = BOX + ( 2.0 * drandm (0) - 1.0 ) * DBOXMX
            RATBOX = BOX / BOXNEW
            RRBOX  = 1.0 / RATBOX
            RCUTN  = RCUT * RRBOX

C       ** CALCULATE SCALING PARAMETERS **

            RAT6   = RATBOX ** 6
            RAT12  = RAT6 * RAT6

C       ** SCALE ENERGY, AND VIRIAL INCLUDING LRC **

            V12NEW = V12  * RAT12
            V6NEW  = V6   * RAT6
            W12NEW = W12  * RAT12
            W6NEW  = W6   * RAT6

C       ** CALCULATE CHANGE IN ENERGY AND VOLUME **

            DELTV  = V12NEW + V6NEW - V12 - V6
            DPV    = PRESUR * ( BOXNEW ** 3 - VOL )
            DVOL   = 3.0 * TEMP * NOPA  * ALOG ( RATBOX )
            DELTHB = BETA * ( DELTV + DPV + DVOL )

C       ** CHECK FOR ACCEPTANCE **

            IF ( DELTHB .LT. 75.0 ) THEN

               IF ( DELTHB .LE. 0.0 ) THEN

                  V12    = V12NEW
```

```
                  V6     = V6NEW
                  W12    = W12NEW
                  W6     = W6NEW

                 DO 98 I = 1, NOPA

                     RX(I) = RX(I) * RRBOX
                     RY(I) = RY(I) * RRBOX
                     RZ(I) = RZ(I) * RRBOX

98                 CONTINUE


                  BOX    = BOXNEW
                  RCUT   = RCUTN
                  ACBOXA = ACBOXA + 1
                  BOXINV = 1.0 / BOX
                  VOL    = BOX ** 3
                  DENS   = NOPA / VOL

               ELSEIF ( EXP ( - DELTHB ) .GT. drandm (0) )THEN

                  V12    = V12NEW
                  V6     = V6NEW
                  W12    = W12NEW
                  W6     = W6NEW

                  DO 99 I = 1, NOPA

                     RX(I) = RX(I) * RRBOX
                     RY(I) = RY(I) * RRBOX
                     RZ(I) = RZ(I) * RRBOX

99                 CONTINUE


                  BOX    = BOXNEW
                  RCUT  = RCUTN
                  ACBOXA = ACBOXA + 1
                  BOXINV = 1.0 / BOX
                  VOL    = BOX ** 3
                  DENS   = NOPA / VOL


               ENDIF

            ENDIF


C       ** ENDS ATTEMPTED BOX MOVE **

          endif

 50         CONTINUE

C       Calculate properties at the end of the step.
           VN   = ( V12 + V6 ) / NOPA
           PRES = DENS * TEMP +  ( W12 + W6 )  / VOL
           h = VN + PRESUR/DENS

C       ** INCREMENT ACCUMULATORS **

           ACM = ACM + 1
```

```
               ACV = ACV + VN
               ACP = ACP + PRES
               ACD = ACD + DENS
               ach = ach + h

               acx(1) = acx(1) + x_a(1)
               acx(2) = acx(2) + x_a(2)

               ACVSQ = ACVSQ + VN ** 2
               ACPSQ = ACPSQ + PRES ** 2
               ACDSQ = ACDSQ + DENS ** 2

C       ** PERFORM PERIODIC OPERATIONS **

          IF ( MOD ( STEP, IRATIO ) .EQ. 0 ) THEN

C          ** ADJUST MAXIMUM DISPLACEMENT FOR ATOMS **

               RATIO = REAL(ACATMA) / REAL(dcount)

               IF ( RATIO .GT. 0.5 ) THEN
                  DRMAX = DRMAX * 1.05
               ELSE
                  DRMAX = DRMAX * 0.95
               ENDIF

               ACATMA = 0
               dcount = 0

               sratio = REAL(ACSWAP)/REAL(scount)
               ACSWAP = 0
               scount = 0

           ENDIF

          IF ( MOD ( STEP, IRATB ) .EQ. 0 ) THEN

C          ** ADJUST MAXIMUM DISPLACEMENT FOR THE BOX **

               BRATIO = REAL(ACBOXA)/REAL(vcount)

               IF ( BRATIO .GT. 0.5 ) THEN
                  DBOXMX = DBOXMX * 1.05
               ELSE
                  DBOXMX = DBOXMX * 0.95
               ENDIF

               ACBOXA = 0
               vcount = 0
           ENDIF


          if (SFLAG .eq. 1) then
           IF ( MOD ( STEP, IPRINT ) .EQ. 0 ) THEN

C          ** OPTIONALLY PRINT INFORMATION **

c             WRITE(4,'(1X,I9,7(1X,F10.5))')
c     :              STEP, VN, PRES, DENS,x_a(2),RATIO,BRATIO, SRATIO
c            write(54,*) STEP,x_a(2)
c            write(64,*) STEP,PRES
c            write(74,*) STEP,DENS
```

```
c           CNFILE = 'a_phase.xyz'
c           CALL writcn_a ( CNFILE )

           ENDIF
          endif

c          if (SFLAG .eq. 3) then
c       Calculate block averages.

          IF ( MOD ( STEP, IBLOCK) .EQ. 0) THEN

              NORM = REAL ( ACM )
              AVV  = ACV / NORM
              AVP  = ACP / NORM
              AVD  = ACD / NORM
              avh  = ach/ NORM

              do  k=1,NCOMPA
                 avx(k) = acx(k)/NORM
              enddo

c             calculate average volume.
              avVOL = NOPA/AVD

c       Write block averages.
c              WRITE(44,10006) STEP,avx(2),AVD,AVP,avh

c       Summations for calculation of mean and std deviation of block
c       average data.

              AVV_SUM   = AVV_SUM   + AVV
              AVP_SUM   = AVP_SUM   + AVP
              AVD_SUM   = AVD_SUM   + AVD
              avh_SUM   = avh_SUM   + avh
              avx2_SUM  = avx2_SUM  + avx(2)
              avVOL_SUM = avVOL_SUM + avVOL

              SQV  = AVV*AVV
              SQP  = AVP*AVP
              SQD  = AVD*AVD
              SQH  = AVH*AVH
              SQX2 =  avx(2)* avx(2)

              SQV_SUM  = SQV_SUM  + SQV
              SQP_SUM  = SQP_SUM  + SQP
              SQD_SUM  = SQD_SUM  + SQD
              SQH_SUM  = SQH_SUM  + SQH
              SQX2_SUM = SQX2_SUM + SQX2

c             only reset accumulators if not in corrector segment.
              if (j.le.1) then
                 ACM    = 0
                 ACATMA = 0
                 ACBOXA = 0

                 ACV = 0.0
                 ACP = 0.0
                 ACD = 0.0
                 ach = 0.0

                 ACVSQ = 0.0
                 ACPSQ = 0.0
```

```
                ACDSQ = 0.0

                FLV = 0.0
                FLP = 0.0
                FLD = 0.0

                dcount = 0
                vcount = 0

                ACSWAP = 0

                do  k=1,NCOMPA
                   acx(k) = 0.0
                enddo

                scount = 0

             end if

          ENDIF
c       endif


 100     CONTINUE

C    ********************************************************************
C    ** MAIN LOOP ENDS                                                 **
C    ********************************************************************

c        WRITE(4,'(/1X,''**** END OF MARKOV CHAIN **** ''//)')

c   Calculate final property averages and standard deviations
        NOBS = real(nstep)/real(iblock)

        IF (NOBS.GT.1.0) THEN

        AVV    = AVV_SUM/NOBS
        AVP    = AVP_SUM/NOBS
        AVD    = AVD_SUM/NOBS
        avh    = avh_SUM/NOBS
        avx(2) = avx2_SUM/NOBS
        avVOL  = avVOL_SUM/NOBS
C        write(99,*)
C        write(99,*)
        STDV  = ABS(NOBS*SQV_SUM-AVV_SUM*AVV_SUM)/(NOBS*(NOBS-1))
C        write(99,*) STDV
        STDV  = SQRT(STDV)
        STDP  = ABS(NOBS*SQP_SUM-AVP_SUM*AVP_SUM)/(NOBS*(NOBS-1))
C        write(99,*) STDP
        STDP  = SQRT(STDP)
        STDD  = ABS(NOBS*SQD_SUM-AVD_SUM*AVD_SUM)/(NOBS*(NOBS-1))
C        write(99,*) STDD
        STDD  = SQRT(STDD)
        STDH  = ABS(NOBS*SQH_SUM-avh_SUM*avh_SUM)/(NOBS*(NOBS-1))
C        write(99,*) STDH
        STDH  = SQRT(STDH)
        STDX2 = ABS(NOBS*SQX2_SUM-avx2_SUM*avx2_SUM)/(NOBS*(NOBS-1))
C        write(99,*) STDX2
        STDX2 = SQRT(STDX2)

        ENDIF

C    ** WRITE OUT FINAL CONFIGURATION AND BOXLENGTH **
```

```
        CNFILE = 'a_phase.xyz'
        CALL writcn_a ( CNFILE )

C    ** WRITE OUT FINAL AVERAGES **

c        NORM = REAL ( ACM )
c        AVV  = ACV / NORM
c        AVP  = ACP / NORM
c        AVD  = ACD / NORM
c        avh  = ach/ NORM

c        do  k=1,NCOMPA
c         avx(k) = acx(k)/NORM
c        enddo

c      calculate average volume.
c        avVOL = NOPA/AVD

c        ACVSQ = ( ACVSQ / NORM ) - AVV ** 2
c        ACPSQ = ( ACPSQ / NORM ) - AVP ** 2
c        ACDSQ = ( ACDSQ / NORM ) - AVD ** 2

c     calculate excess heat capacity
c        CPX_a = ACVSQ/(TEMP*TEMP)

c        IF ( ACVSQ .GT. 0.0 ) FLV = SQRT ( ACVSQ )
c        IF ( ACPSQ .GT. 0.0 ) FLP = SQRT ( ACPSQ )
c        IF ( ACDSQ .GT. 0.0 ) FLD = SQRT ( ACDSQ )



c        WRITE(4, 10004)
c        WRITE(4,'('' AVERAGES'',5(2X,F10.4))') AVV,AVP,AVD,avh,avx(2)
c        WRITE(4,'('' FLUCTS  '',3(2X,F10.5))') FLV, FLP, FLD

c        close(unit=44)
c       close(unit=54)
c       close(unit=64)
c       close(unit=74)



10001   FORMAT(//1X,'  STEP    ..POTENT..  ..PRESSURE.. ..DENSITY..',
     :              '  ..RATIO..  ..BRATIO..')
10002   FORMAT(//1X,'          ..POTENT..  ..PRESSURE.. ..DENSITY..',
     :              '  ..enthalpy..')
10003   FORMAT(//1X,'  STEP    ..POTENT..  ..PRESSURE.. ..DENSITY..',
     :              '..x_a(2)..  ..RATIO..  ..BRATIO.. ..SRATIO..')
10004   FORMAT(//1X,'          ..POTENT..  ..PRESSURE.. ..DENSITY..',
     :              '  ..enthalpy..  ..x_a(2)..')
10005   FORMAT(1X,'STEP',4X,'x_2',5X,'dens',4X,'Pres',5X,'h')
10006   FORMAT(I7,2X,F6.4,2X,F6.4,2X,F7.4,2X,F7.4)

c         STOP
        return

        END
```

```
      subroutine mcnpt_b(nstep,infile,outfile,bfile,j,SFLAG)

c August 30, 1999
c MRH  corrections are in lower case.
c A&T original code in upper case.

c Changed random number generator to  drandm (9/10/96).
c Make choice of mc move with a certain probability. (7/24/97)
c Select atom to displace at random.                  (7/24/97)
c Select volume change with probability 5/(NOP+5).    (11/26/97)
c Corrected mistake in longrange correction virial term
c    term3w, (4/14/99)
c Remove CPP directives. (5/24/99)
c Sample properties at the end of each step. (5/24/99)

c23456789012345678901234567890123456789012345678901234567890123456789012
C     ******************************************************************
C     ** THIS FORTRAN CODE IS INTENDED TO ILLUSTRATE POINTS MADE IN    **
C     ** THE TEXT. TO OUR KNOWLEDGE IT WORKS CORRECTLY. HOWEVER IT IS  **
C     ** THE RESPONSIBILITY OF THE USER TO TEST IT, IF IT IS USED IN A **
C     ** RESEARCH APPLICATION.                                         **
C     ******************************************************************

C     ******************************************************************
C     ** FICHE F.12                                                    **
C     ** MONTE CARLO SIMULATION IN THE CONSTANT-NPT ENSEMBLE.          **
C     ******************************************************************

C     ******************************************************************
C     ** MONTE CARLO SIMULATION IN THE CONSTANT-NPT ENSEMBLE.          **
C     **                                                              **
C     ** THIS PROGRAM TAKES A CONFIGURATION OF LENNARD JONES ATOMS     **
C     ** AND PERFORMS A MONTE CARLO SIMULATION AT CONSTANT NPT. THE    **
C     ** BOX IS IN UNITS OF SIGMA THE LENNARD JONES DIAMETER.          **
C     ** THERE ARE NO LOOKUP TABLES INCLUDED.                          **
C     **                                                              **
C     ** REFERENCE:                                                   **
C     **                                                              **
C     ** MCDONALD, CHEM. PHYS. LETT. 3, 241, 1969.                    **
C     **                                                              **
C     ** PRINCIPAL VARIABLES:                                         **
C     **                                                              **
C     ** INTEGER N                          NUMBER OF MOLECULES       **
C     ** REAL    RX(N),RY(N),RZ(N)          POSITIONS                 **
C     ** REAL    VOL                        VOLUME                    **
C     ** REAL    BOX                        BOX LENGTH                **
C     ** REAL    DENS                       REDUCED DENSITY           **
C     ** REAL    TEMP                       REDUCED TEMPERATURE       **
C     ** REAL    SIGMA                      REDUCED LJ DIAMETER       **
C     ** REAL    DRMAX                      MAXIMUM DISPLACEMENT      **
C     ** REAL    V                          THE POTENTIAL ENERGY      **
C     ** REAL    W                          THE VIRIAL                **
C     ** REAL    PRESUR                     REQUIRED PRESSURE         **
C     ** REAL    DBOXMX                     MAX CHANGE IN BOX         **
c      ** real       h                                 enthalpy
C     **                                                              **
C     ** USAGE:                                                       **
C     **                                                              **
C     ** CONDUCTS MONTE CARLO SIMULATION AT CONSTANT PRESSURE FOR A    **
C     ** SPECIFIED NUMBER OF CYCLES FROM A GIVEN INITIAL CONFIGURATION.**
C     **                                                              **
C     ** UNITS:                                                       **
```

```
C    **                                                              **
C    ** THIS PROGRAM USES THE USUAL REDUCED LJ UNITS. IN PARTICULAR   **
C    ** THE BOX LENGTH IS IN UNITS OF SIGMA.                         **
C    **                                                              **
C    ** ROUTINES REFERENCED:                                         **
C    **                                                              **
C    ** SUBROUTINE ENERGY ( RXI, RYI, RZI, I, RCUT, BOX, V12, V6,    **
C    **    :                 W12, W6 )                               **
C    **    CALCULATES THE ENERGY AND VIRIAL FOR ATOM I IN THE FLUID   **
C    ** SUBROUTINE READCN ( CNFILE, BOX )                            **
C    **    READS IN CONFIGURATION AND BOX VARIABLES                  **
C    ** SUBROUTINE SUMUP ( RCUT, RMIN, OVRLAP, BOX, V12, V6, W12, W6 )**
C    **    CALCULATES POTENTIAL AND VIRIAL FOR A CONFIGURATION       **
C    ** SUBROUTINE WRITCN ( CNFILE, BOX )                            **
C    **    WRITES OUT CONFIGURATION AND BOX VARIABLES                **
C    ******************************************************************



        REAL        RX(NOPB), RY(NOPB), RZ(NOPB)

        INTEGER     STEP, NSTEP, IPRINT, IBLOCK,IRATIO, IRATB, I
        INTEGER     ILOOP,rnd_num
        integer     dcount,vcount
        integer     ACM,ACATMA,ACBOXA
        double precision ACV, ACP, ACD, NORM
        double precision ach,ACVSQ, ACPSQ, ACDSQ
        REAL        AVV, AVP, AVD
        REAL        FLV, FLP, FLD
        REAL        DENS, TEMP, RCUT, RMIN, PRESUR, VOL, PRES, VN
        REAL        BOXINV, BOXNEW, RATBOX, RAT12, RAT6, DVOL, DPV
        REAL        DELTHB, DRMAX, DBOXMX, BETA, DUMMY, RATIO
        REAL        RRBOX, BRATIO, DELTVB, RCUTN
        REAL        RXIOLD, RYIOLD, RZIOLD, RXINEW, RYINEW, RZINEW
        real        RXINEW2, RYINEW2, RZINEW2
        REAL        V12OLD, V6OLD, V12NEW, V6NEW, VS
        REAL        W12OLD, W6OLD, W12NEW, W6NEW, WS, PS
        REAL        DELV12, DELV6, DELW12, DELW6, DELTV
        REAL        V6, W6, V12, W12, VLRC, VLRCN, WLRC, WLRCN
        REAL        SR3, SR9, VLRC6, WLRC6, VLRC12, WLRC12, PI
        real           h,CPX_b
#ifdef SLE
        real       cell_con
#endif

        real       avVOL,BOX,avh

        integer     j,ii,jj
        INTEGER IMOVE,NMOVE,NSWAP, SFLAG

        CHARACTER   TITLE*80, CNFILE*30
        character   infile*30, outfile*30, bfile*30
         character   xfile*30,pfile*30,dfile*30
        LOGICAL     OVRLAP

        real NOBS
        real SQV,SQP,SQD,SQH,SQX2
        real SQV_SUM,SQP_SUM,SQD_SUM,SQH_SUM,SQX2_SUM
        real AVV_SUM,AVP_SUM,AVD_SUM,avh_SUM,avx2_SUM,avVOL_SUM
        real STDV,STDP,STDD,STDH,STDX2

c       semigd variables
```

```
        integer species(NOPB)
        integer N_b(NCOMPB),Ntry(NCOMPB)
        real    sigmas_b(NCOMPB,NCOMPB),eps_b(NCOMPB,NCOMPB)
        real    sig2_b(NCOMPB,NCOMPB),sig3_b(NCOMPB,NCOMPB)
        real    eps_b4(NCOMPB,NCOMPB)
        real  RCUT3
        real term3u
        real term3w
        real term9u
        real term9w
        integer ACSWAP,scount,tryiden
        double precision acx(NCOMPB)
        real x_b(NCOMPB),avx(NCOMPB)
        real  sratio,m,xi2
        real xnew(NCOMPB)


c       end of semigd variables

        PARAMETER ( PI = 3.1415927 )


        COMMON / config_b / RX, RY, RZ
        common / syspar / TEMP,PRESUR

#ifdef SLE
        common / block14 / avVOL,BOX,avh,cell_con
#else
        common / block14 / avVOL,BOX,avh
#endif

         common / binb1 / sigmas_b,eps_b
         common / binb2 / sig2_b,sig3_b,eps_b4
         common / block12 / species,N_b
         common / block19 / x_b,avx,STDX2
         common / fugacity / xi2
         common / swap /  NSWAP
         common / block20 / xfile,pfile,dfile


external time,drandm,srand,rand
        real*8 drandm
        integer*4 iseed
        integer time


c For random number generator
c        iseed = 4
        iseed = time()
d = drandm(iseed)
        call srand(iseed)



C    ***********************************************************************

C    ** BASIC SIMULATION PARAMETERS **

         open ( unit = 8, file = infile, status = 'old')

c       RUN TITLE
        READ (8,'(/A7)') TITLE
c       INTERVAL BETWEEN PRINTS IN CYCLES
```

```
        READ (8,'(/I7)') IPRINT
c       INTERVAL FOR BLOCK AVERAGES
         READ (8,'(/I7)') IBLOCK
c       INTERVAL FOR UPDATE OF MAXIMUM DISPLACEMENT OF ATOMS IN CYCLES
        READ (8,'(/I7)') IRATIO
c       INTERVAL FOR UPDATE OF MAXIMUM DISPLACEMENT OF THE BOX IN CYCLES
        READ (8,'(/I7)') IRATB

        close (unit = 8)

c         if (SFLAG .eq. 1) then
c         open ( unit = 54, file = xfile, status = 'unknown')
c         open ( unit = 64, file = pfile, status = 'unknown')
c         open ( unit = 74, file = dfile, status = 'unknown')
c         endif


c         open ( unit = 4, file = outfile, status = 'unknown')
c         open ( unit = 44, file = bfile, status = 'unknown')
c         write(44,*) 'Block Averages'
c         write(44,10005)

c        write(4,'('' Update = '',I3)') j
c        write(4,*)


c       WRITE(4,'(1H1,'' **** PROGRAM MCSEMI ****              '')')
c       WRITE(4,'(//''MONTE CARLO IN A CONSTANT-N,P,T,xi2 ENSEMBLE  '')')


c        WRITE(4,'(  '' LENNARD-JONES ATOMS                      '')')


c        WRITE(4,'(//1X,A)') TITLE
c        WRITE(4,'('' NUMBER OF CYCLES                = '',I9)') NSTEP
c        WRITE(4,'('' PRINT INTERVAL                  = '',I9)') IPRINT
c        WRITE(4,'('' RATIO UPDATE INTERVAL FOR ATOMS = '',I9)') IRATIO
c        WRITE(4,'('' RATIO UPDATE INTERVAL FOR BOX   = '',I9)') IRATB
c        WRITE(4,'('' DESIRED PRES.                   = '',F10.5)')PRESUR
c        WRITE(4,'('' DESIRED TEMP                    = '',F10.5)')TEMP
c        WRITE(4,'('' Box length                      = '',F10.5)')BOX

c        WRITE(4,'('' Fugacity fraction (species 2)   = '',F10.5)')xi2
c        WRITE(4,'('' Mole fraction (species 2)       = '',F10.5)')x_b(2)

C    ** SET DEPENDENT PARAMETERS **

        VOL    = BOX*BOX*BOX
        BOXINV = 1.0 / BOX
        DENS   = REAL ( NOPB ) / VOL
        RCUT   = 0.5*box
c        RCUT=2.5*sigma

        IF ( RCUT .GT. ( 0.5 * BOX ) ) STOP 'CUT-OFF TOO LARGE'

        DBOXMX = BOX / 40.0

#ifdef SLE
c sample over range(-cell/4,cell/4)
        drmax = cell_con/2.0
#else
c drmax = box/1.5
        DRMAX  = 0.15
```

```
#endif

        RMIN   = 0.70
        BETA   = 1.0 / TEMP

c        WRITE(4,'('' INITIAL DENSITY                = '',F10.5)') DENS

C    ** CALCULATE LONG-RANGE CORRECTIONS FOR LJ POTENTIAL.    **
C    ** 6 IS FOR ATTRACTIVE CONTRIBUTIONS 12 IS FOR REPULSIVE **



      term3u      = 0.0
      term9u      = 0.0
      term3w      = 0.0
      term9w      = 0.0
      VLRC12      = 0.0
      VLRC6       = 0.0
      WLRC12      = 0.0
      WLRC6       = 0.0

      RCUT3       = RCUT*RCUT*RCUT

      do  ii=1,NCOMPB
         do  jj=1,NCOMPB
            SR3 = sig3_b(ii,jj)/RCUT3
            SR9 = SR3*SR3*SR3

            term9u =  eps_b(ii,jj)* sig3_b(ii,jj)* SR9/9.
            term3u =  eps_b(ii,jj)* sig3_b(ii,jj)*(-SR3)/3.
            term9w = -eps_b(ii,jj)* sig3_b(ii,jj)* SR9*4./3.
            term3w = -eps_b(ii,jj)* sig3_b(ii,jj)*(-2.)*SR3

            VLRC12 = term9u * x_b(ii)* x_b(jj) + VLRC12
            VLRC6  = term3u * x_b(ii)* x_b(jj) + VLRC6
            WLRC12 = term9w * x_b(ii)* x_b(jj) + WLRC12
            WLRC6  = term3w * x_b(ii)* x_b(jj) + WLRC6

         enddo
      enddo

      VLRC12      =  2.* PI* DENS* NOPB* VLRC12
      VLRC6       =  2.* PI* DENS* NOPB* VLRC6
      WLRC12      = -2.* PI* DENS* NOPB* WLRC12/3.
      WLRC6       = -2.* PI* DENS* NOPB* WLRC6/3.

        VLRC   =    VLRC12 + VLRC6
        WLRC   =    WLRC12 + WLRC6



C    ** ZERO ACCUMULATORS **

        ratio = 0.0
        sratio = 0.0
        bratio = 0.0

        if (j.le.1) then
        ACM    = 0
        ACATMA = 0
        ACBOXA = 0

        ACV = 0.0
```

```
         ACP = 0.0
         ACD = 0.0
         ach = 0.0

         ACVSQ = 0.0
         ACPSQ = 0.0
         ACDSQ = 0.0

         FLV = 0.0
         FLP = 0.0
         FLD = 0.0

         dcount = 0
         vcount = 0

         ACSWAP = 0

         do  k=1,NCOMPB
          acx(k) = 0.0
         enddo

         scount = 0

         end if

         AVV_SUM = 0.0
         AVP_SUM = 0.0
         AVD_SUM = 0.0
         avh_SUM = 0.0
         avx2_SUM = 0.0
         avVOL_SUM = 0.0

         SQV_SUM = 0.0
         SQP_SUM = 0.0
         SQD_SUM = 0.0
         SQH_SUM = 0.0
         SQX2_SUM = 0.0

C    ** CALCULATE INITIAL ENERGY AND VIRIAL **

         call sumup_b ( RCUT, RMIN, OVRLAP, V12, V6, W12, W6 )

c         IF ( OVRLAP ) STOP 'OVERLAP IN INITIAL CONFIGURATION'

C    ** CALCULATE THE INITIAL ENERGY AND VIRIAL **

         VS = ( V12 + V6 + VLRC ) / REAL ( NOPB )
         WS = ( W12 + W6 + WLRC ) / REAL ( NOPB )
         PS = DENS * TEMP + ( W12 + W6 + WLRC ) / VOL

C    ** ADD LONG RANGE CORRECTIONS **
C    ** INTO THE ENERGY AND VIRIAL **

c         write(99,*) 'energy and virial w/out correction'
c         write(99,*) 'V12', V12
c         write(99,*) 'V6', V6
c         write(99,*) 'W12',W12
c         write(99,*) 'W6',W6


         V12 = V12 + VLRC12
         V6  = V6  + VLRC6
         W12 = W12 + WLRC12
```

```
        W6  = W6  + WLRC6

c        write(99,*) 'lrange correction'
c        write(99,*) 'VLRC12', VLRC12
c        write(99,*) 'VLRC6', VLRC6
c        write(99,*) 'WLRC12',WLRC12
c        write(99,*) 'WLRC6',WLRC6


c        WRITE(4,'('' INITIAL V/N                    = '',F10.6)') VS
c        WRITE(4,'('' INITIAL W/N                    = '',F10.6)') WS
c        WRITE(4,'('' INITIAL P                      = '',F10.6)') PS
c        WRITE(4,'(//'' **** START OF MARKOV CHAIN ****'')')

c        WRITE(4,10003)


c         NMOVE = 2*NOPB+1
          NMOVE = NOPB + NSWAP + 1

C    *********************************************************************
C    ** MAIN LOOP STARTS                                                **
C    *********************************************************************

        DO 100 STEP = 1, NSTEP

            DO 50 IMOVE = 1,NMOVE

            rnd_num=INT( drandm(0)*NMOVE ) + 1

            if(rnd_num.le.NOPB) then
c           ** trial particle displacement **

              dcount = dcount + 1

c         select atom to move
c               I = rnd_num
                I =INT( drandm(0)*(NOPB-1) ) + 1

               RXIOLD = RX(I)
               RYIOLD = RY(I)
               RZIOLD = RZ(I)

C          ** CALCULATE V FOR AN ATOM IN OLD STATE **

               call energy_b ( RXIOLD, RYIOLD, RZIOLD, I, RCUT,
     :                         V12OLD, V6OLD, W12OLD, W6OLD )

C          ** MOVE ATOM I **
c          leave new positions in uncorrected form

 90            RXINEW = RXIOLD + ( 2.0 * drandm(0) - 1.0 ) * DRMAX
               RYINEW = RYIOLD + ( 2.0 * drandm(0) - 1.0 ) * DRMAX
               RZINEW = RZIOLD + ( 2.0 * drandm(0) - 1.0 ) * DRMAX

c          correct positions to stay in central box
               RXINEW = RXINEW - ANINT ( RXINEW * BOXINV ) * BOX
               RYINEW = RYINEW - ANINT ( RYINEW * BOXINV ) * BOX
               RZINEW = RZINEW - ANINT ( RZINEW * BOXINV ) * BOX


C          ** CALCULATE V FOR ATOM IN NEW STATE **
```

```
               call energy_b ( RXINEW, RYINEW, RZINEW, I, RCUT,
     :                         V12NEW, V6NEW, W12NEW, W6NEW )

c if(ovrlap) goto 90

C          ** CHECK FOR ACCEPTANCE **

               DELV12 = V12NEW - V12OLD
               DELV6  = V6NEW  - V6OLD
               DELW12 = W12NEW - W12OLD
               DELW6  = W6NEW  - W6OLD
               DELTV  = DELV12 + DELV6
               DELTVB = BETA * DELTV

               IF ( DELTVB .LT. 75.0 ) THEN

                  IF ( DELTV .LE. 0.0 ) THEN

                     V12    = V12 + DELV12
                     V6     = V6  + DELV6
                     W12    = W12 + DELW12
                     W6     = W6  + DELW6
                     RX(I)  = RXINEW
                     RY(I)  = RYINEW
                     RZ(I)  = RZINEW
                     ACATMA = ACATMA + 1

                  ELSEIF ( EXP ( - DELTVB ) .GT. drandm (0) ) THEN
c              IF ( EXP ( - DELTVB ) .GT. drandm (0) ) THEN

                     V12    = V12 + DELV12
                     V6     = V6  + DELV6
                     W12    = W12 + DELW12
                     W6     = W6  + DELW6
                     RX(I)  = RXINEW
                     RY(I)  = RYINEW
                     RZ(I)  = RZINEW
                     ACATMA = ACATMA + 1

                  ENDIF

               ENDIF

C       ** ENDS ATOMS DISPLACEMENT STEP   **

c           elseif(rnd_num.gt.NOPB .and. rnd_num.le.2*NOPB) then
            elseif(rnd_num.gt.NOPB .and. rnd_num.le.(NOPB+NSWAP)) then
c       ** attempt particle identity change **

               scount = scount + 1
c       ** select particle **
c               I = rnd_num - NOPB
                I =INT( drandm(0)*(NOPB-1) ) + 1

               RXIOLD = RX(I)
               RYIOLD = RY(I)
               RZIOLD = RZ(I)

C          ** CALCULATE V FOR A particle IN OLD STATE **

               call energy_b ( RXIOLD, RYIOLD, RZIOLD, I, RCUT,
     :                         V12OLD, V6OLD, W12OLD, W6OLD )
```

```
c          ** switch identity of particle **
c              write(99,*) 'particle', I,species(I),N_b
               if(species(I).eq.1) then
                  tryiden = 2
                  m       = 1.0
                  Ntry(2) = N_b(2) + 1
                  Ntry(1) = N_b(1) - 1
                  xnew(2) = REAL(Ntry(2))/NOPB
                  xnew(1) = REAL(Ntry(1)) /NOPB
               else
                  tryiden = 1
                  m       =-1.0
                  Ntry(1) = N_b(1) + 1
                  Ntry(2) = N_b(2) - 1
                  xnew(1) = REAL(Ntry(1))/NOPB
                  xnew(2) = REAL(Ntry(2)) /NOPB

               endif

C          ** CALCULATE V FOR A particle IN NEW STATE **

               call energi_b ( RXIOLD, RYIOLD, RZIOLD, I, RCUT,
     :                         V12NEW, V6NEW, W12NEW, W6NEW,tryiden )

c          ** Calculate difference in long ranges corrections **
               term3u      = 0.0
               term9u      = 0.0
               term3w      = 0.0
               term9w      = 0.0
               VLRC12      = 0.0
               VLRC6       = 0.0
               WLRC12      = 0.0
               WLRC6       = 0.0

               RCUT3       = RCUT*RCUT*RCUT

               do  ii=1,NCOMPB
                  do  jj=1,NCOMPB
                     SR3 = sig3_b(ii,jj)/RCUT3
                     SR9 = SR3*SR3*SR3

                     term9u =  eps_b(ii,jj)* sig3_b(ii,jj)* SR9/9.
                     term3u =  eps_b(ii,jj)* sig3_b(ii,jj)*(-SR3)/3.
                     term9w = -eps_b(ii,jj)* sig3_b(ii,jj)* SR9*4./3.
                     term3w = -eps_b(ii,jj)* sig3_b(ii,jj)*(-2.)*SR3

            VLRC12 = term9u *(xnew(ii)*xnew(jj)-x_b(ii)*x_b(jj)) + VLRC12
            VLRC6  = term3u *(xnew(ii)*xnew(jj)-x_b(ii)*x_b(jj)) + VLRC6
            WLRC12 = term9w *(xnew(ii)*xnew(jj)-x_b(ii)*x_b(jj)) + WLRC12
            WLRC6  = term3w *(xnew(ii)*xnew(jj)-x_b(ii)*x_b(jj)) + WLRC6

                  enddo
               enddo

c           ** difference between LRC in old and new state **
               VLRC12      =  2.* PI* DENS* NOPB* VLRC12
               VLRC6       =  2.* PI* DENS* NOPB* VLRC6
               WLRC12      = -2.* PI* DENS* NOPB* WLRC12/3.
               WLRC6       = -2.* PI* DENS* NOPB* WLRC6/3.

C          ** CHECK FOR ACCEPTANCE **

               DELV12 = V12NEW - V12OLD + VLRC12
```

```
               DELV6  = V6NEW  - V6OLD + VLRC6
               DELW12 = W12NEW - W12OLD + WLRC12
               DELW6  = W6NEW  - W6OLD + WLRC6
               DELTV  = DELV12 + DELV6
               DELTVB = BETA * DELTV - m*ALOG(xi2/(1.-xi2))

               IF ( DELTVB .LT. 75.0 ) THEN

                  IF ( DELTVB .LE. 0.0 ) THEN

                     V12    = V12 + DELV12
                     V6     = V6  + DELV6
                     W12    = W12 + DELW12
                     W6     = W6  + DELW6
                     species(I) = tryiden
                     x_b(1) = xnew(1)
                     x_b(2) = xnew(2)
                     N_b(1) = Ntry(1)
                     N_b(2) = Ntry(2)
                     ACSWAP  = ACSWAP + 1

                  ELSEIF ( EXP ( - DELTVB ) .GT. drandm(0) ) THEN

                     V12    = V12 + DELV12
                     V6     = V6  + DELV6
                     W12    = W12 + DELW12
                     W6     = W6  + DELW6
                     species(I) = tryiden
                     x_b(1) = xnew(1)
                     x_b(2) = xnew(2)
                     N_b(1) = Ntry(1)
                     N_b(2) = Ntry(2)
                     ACSWAP  = ACSWAP + 1

                  ENDIF

               ENDIF

C       ** ENDS particle identity change STEP   **

            else

                vcount = vcount + 1
C       ** ATTEMPT A BOX MOVE **

              BOXNEW = BOX + ( 2.0 * drandm (0) - 1.0 ) * DBOXMX
               RATBOX = BOX / BOXNEW
               RRBOX  = 1.0 / RATBOX
               RCUTN  = RCUT * RRBOX

C       ** CALCULATE SCALING PARAMETERS **

               RAT6   = RATBOX ** 6
               RAT12  = RAT6 * RAT6

C       ** SCALE ENERGY, AND VIRIAL INCLUDING LRC **

               V12NEW = V12  * RAT12
               V6NEW  = V6   * RAT6
               W12NEW = W12  * RAT12
               W6NEW  = W6   * RAT6

C       ** CALCULATE CHANGE IN ENERGY AND VOLUME **
```

```
            DELTV  = V12NEW + V6NEW - V12 - V6
            DPV    = PRESUR * ( BOXNEW ** 3 - VOL )
            DVOL   = 3.0 * TEMP * REAL ( NOPB ) * ALOG ( RATBOX )
            DELTHB = BETA * ( DELTV + DPV + DVOL )

C       ** CHECK FOR ACCEPTANCE **

            IF ( DELTHB .LT. 75.0 ) THEN

               IF ( DELTHB .LE. 0.0 ) THEN

                 V12    = V12NEW
                  V6     = V6NEW
                  W12    = W12NEW
                  W6     = W6NEW

                 DO 98 I = 1, NOPB

                     RX(I) = RX(I) * RRBOX
                     RY(I) = RY(I) * RRBOX
                     RZ(I) = RZ(I) * RRBOX

 98                CONTINUE

#ifdef SLE
                  cell_con=cell_con*rrbox
 drmax=drmax*rrbox
#endif

                  BOX    = BOXNEW
                  RCUT   = RCUTN
                  ACBOXA = ACBOXA + 1
                  BOXINV = 1.0 / BOX
                  VOL    = BOX ** 3
                  DENS   = REAL ( NOPB ) / VOL

               ELSEIF ( EXP ( - DELTHB ) .GT. drandm (0) )THEN
c              IF ( EXP ( - DELTHB ) .GT. drandm (0) )THEN
                  V12    = V12NEW
                  V6     = V6NEW
                  W12    = W12NEW
                  W6     = W6NEW

                  DO 99 I = 1, NOPB

                     RX(I) = RX(I) * RRBOX
                     RY(I) = RY(I) * RRBOX
                     RZ(I) = RZ(I) * RRBOX

 99                CONTINUE

#ifdef SLE
                  cell_con=cell_con*rrbox
 drmax=drmax*rrbox
#endif


                  BOX    = BOXNEW
                  RCUT  = RCUTN
                  ACBOXA = ACBOXA + 1
                  BOXINV = 1.0 / BOX
                  VOL    = BOX ** 3
```

```
                DENS   = REAL ( NOPB ) / VOL

              ENDIF

            ENDIF


C       ** ENDS ATTEMPTED BOX MOVE **

         endif

 50      CONTINUE

C       Calculate properties at the end of the step.
              VN   = ( V12 + V6 ) / REAL ( NOPB )
              PRES = DENS * TEMP +  ( W12 + W6 )  / VOL
              h = VN + PRESUR/DENS

C       ** INCREMENT ACCUMULATORS **

              ACM = ACM + 1
              ACV = ACV + VN
              ACP = ACP + PRES
              ACD = ACD + DENS
              ach = ach + h

              acx(1) = acx(1) + x_b(1)
              acx(2) = acx(2) + x_b(2)


              ACVSQ = ACVSQ + VN ** 2
              ACPSQ = ACPSQ + PRES ** 2
              ACDSQ = ACDSQ + DENS ** 2


C       ** PERFORM PERIODIC OPERATIONS **


           IF ( MOD ( STEP, IRATIO ) .EQ. 0 ) THEN

C           ** ADJUST MAXIMUM DISPLACEMENT FOR ATOMS **

              RATIO = REAL(ACATMA) / REAL(dcount)

               IF ( RATIO .GT. 0.5 ) THEN

                  DRMAX = DRMAX * 1.05

              ELSE

                  DRMAX = DRMAX * 0.95

               ENDIF

              ACATMA = 0
              dcount = 0

              sratio = REAL(ACSWAP)/REAL(scount)
              ACSWAP = 0
              scount = 0

            ENDIF
```

```
           IF ( MOD ( STEP, IRATB ) .EQ. 0 ) THEN

C          ** ADJUST MAXIMUM DISPLACEMENT FOR THE BOX **

              BRATIO = REAL(ACBOXA)/REAL(vcount)

              IF ( BRATIO .GT. 0.5 ) THEN

                 DBOXMX = DBOXMX * 1.05

              ELSE

                 DBOXMX = DBOXMX * 0.95

              ENDIF

              ACBOXA = 0
              vcount = 0
           ENDIF

          if (SFLAG .eq. 1) then
           IF ( MOD ( STEP, IPRINT ) .EQ. 0 ) THEN

C          ** OPTIONALLY PRINT INFORMATION **

c             WRITE(4,'(1X,I9,7(1X,F10.5))')
c     :               STEP, VN, PRES, DENS,x_b(2),RATIO,BRATIO, SRATIO
c              write(54,*) STEP,x_b(2)
c             write(64,*) STEP,PRES
c             write(74,*) STEP,DENS


c   Use next line only if we want to check intermediate configurations.
c              CALL writcn_b ( CNFILE )

           ENDIF
          endif

c          if (SFLAG .eq. 3) then
c       Calculate block averages.

           IF ( MOD ( STEP, IBLOCK) .EQ. 0) THEN

              NORM = REAL ( ACM )
              AVV  = ACV / NORM
              AVP  = ACP / NORM
              AVD  = ACD / NORM
              avh  = ach/ NORM

              do  k=1,NCOMPA
                 avx(k) = acx(k)/NORM
              enddo

c             calculate average volume.
              avVOL = NOPB/AVD

c       Write block averages.
c               WRITE(44,10006) STEP,avx(2),AVD,AVP,avh

c       Summations for calculation of mean and std deviation of block
c       average data.

              AVV_SUM   = AVV_SUM   + AVV
```

```
                AVP_SUM   = AVP_SUM   + AVP
                AVD_SUM   = AVD_SUM   + AVD
                avh_SUM   = avh_SUM   + avh
                avx2_SUM  = avx2_SUM  + avx(2)
                avVOL_SUM = avVOL_SUM + avVOL

                SQV  = AVV*AVV
                SQP  = AVP*AVP
                SQD  = AVD*AVD
                SQH  = AVH*AVH
                SQX2 =  avx(2)* avx(2)

                SQV_SUM  = SQV_SUM  + SQV
                SQP_SUM  = SQP_SUM  + SQP
                SQD_SUM  = SQD_SUM  + SQD
                SQH_SUM  = SQH_SUM  + SQH
                SQX2_SUM = SQX2_SUM + SQX2


c               only reset accumulators if not in corrector segment.
                if (j.le.1) then
                   ACM    = 0
                   ACATMA = 0
                   ACBOXA = 0

                   ACV = 0.0
                   ACP = 0.0
                   ACD = 0.0
                   ach = 0.0

                   ACVSQ = 0.0
                   ACPSQ = 0.0
                   ACDSQ = 0.0

                   FLV = 0.0
                   FLP = 0.0
                   FLD = 0.0

                   dcount = 0
                   vcount = 0

                   ACSWAP = 0

                   do  k=1,NCOMPB
                      acx(k) = 0.0
                   enddo

                   scount = 0

                end if

             ENDIF
c           endif

 100        CONTINUE

C       ***********************************************************************
C       ** MAIN LOOP ENDS                                                     **
C       ***********************************************************************

c           WRITE(4,'(/1X,''**** END OF MARKOV CHAIN **** ''//)')

c       Calculate final property averages and standard deviations
```

```
         NOBS = real(nstep)/real(iblock)

         IF (NOBS.GT.1.0) THEN

         AVV    = AVV_SUM/NOBS
         AVP    = AVP_SUM/NOBS
         AVD    = AVD_SUM/NOBS
         avh    = avh_SUM/NOBS
         avx(2) = avx2_SUM/NOBS
         avVOL  = avVOL_SUM/NOBS

         STDV  = ABS(NOBS*SQV_SUM-AVV_SUM*AVV_SUM)/(NOBS*(NOBS-1))
         STDV  = SQRT(STDV)
         STDP  = ABS(NOBS*SQP_SUM-AVP_SUM*AVP_SUM)/(NOBS*(NOBS-1))
         STDP  = SQRT(STDP)
         STDD  = ABS(NOBS*SQD_SUM-AVD_SUM*AVD_SUM)/(NOBS*(NOBS-1))
         STDD  = SQRT(STDD)
         STDH  = ABS(NOBS*SQH_SUM-avh_SUM*avh_SUM)/(NOBS*(NOBS-1))
         STDH  = SQRT(STDH)
         STDX2 = ABS(NOBS*SQX2_SUM-avx2_SUM*avx2_SUM)/(NOBS*(NOBS-1))
         STDX2 = SQRT(STDX2)

         ENDIF

c        write(4,*) 'acm = ', acm
c        write(4,*) 'norm = ', norm

C     ** WRITE OUT FINAL CONFIGURATION AND BOXLENGTH **
         CNFILE = 'b_phase.xyz'
         CALL writcn_b ( CNFILE )

C     ** WRITE OUT FINAL AVERAGES **

c         NORM = REAL ( ACM )
c         AVV  = ACV / NORM
c         AVP  = ACP / NORM
c         AVD  = ACD / NORM
c         avh = ach/ NORM

c       calculate average volume.
c         avVOL = NOPB/AVD

c         do  k=1,NCOMPB
c          avx(k) = acx(k)/NORM
c         enddo

c         ACVSQ = ( ACVSQ / NORM ) - AVV ** 2
c         ACPSQ = ( ACPSQ / NORM ) - AVP ** 2
c         ACDSQ = ( ACDSQ / NORM ) - AVD ** 2

c      calculate excess heat capacity
c         CPX_b = ACVSQ/(TEMP*TEMP)

c         IF ( ACVSQ .GT. 0.0 ) FLV = SQRT ( ACVSQ )
c         IF ( ACPSQ .GT. 0.0 ) FLP = SQRT ( ACPSQ )
c         IF ( ACDSQ .GT. 0.0 ) FLD = SQRT ( ACDSQ )


c         WRITE(4, 10004)
c         WRITE(4,'('' AVERAGES'',5(2X,F10.4))') AVV,AVP,AVD,avh,avx(2)
c         WRITE(4,'('' FLUCTS  '',3(2X,F10.5))') FLV, FLP, FLD

c          close(unit=44)
```

```
c         close(unit=54)
c        close(unit=64)
c        close(unit=74)



10001   FORMAT(//1X,'  STEP    ..POTENT..  ..PRESSURE.. ..DENSITY..',
     :              '  ..RATIO..  ..BRATIO..')
10002   FORMAT(//1X,'          ..POTENT..  ..PRESSURE.. ..DENSITY..',
     :              '  ..enthalpy..')
10003   FORMAT(//1X,'  STEP    ..POTENT..  ..PRESSURE.. ..DENSITY..',
     :              '..x_b(2)..  ..RATIO..  ..BRATIO.. ..SRATIO..')
10004   FORMAT(//1X,'          ..POTENT..  ..PRESSURE.. ..DENSITY..',
     :              '  ..enthalpy..  ..x_b(2)..')
10005   FORMAT(1X,'STEP',4X,'x_2',5X,'dens',4X,'Pres',5X,'h')
10006   FORMAT(I7,2X,F6.4,2X,F6.4,2X,F7.4,2X,F7.4)

c       STOP

        return

        END
```

```
subroutine rd_data(input,npoints,hen2_a,hen2_b,
     :                         hh,maxlimit,minlimit)


# include "mainhead3.f"
c *****************************************************************
c Input parameters to define the system.

      cnunit=10

#ifdef SINA
#ifdef SINB
      open  (unit = cnunit, file = 'system1.inp', status = 'unknown',
     :                                    form='formatted')
#endif
#endif
#ifdef BINA
#ifdef BINB
      open  (unit = cnunit, file = 'system2.inp', status = 'unknown',
     :                                    form='formatted')
#endif
#endif
#ifdef SINA
#ifdef BINB
      open  (unit = cnunit, file = 'system3.inp', status = 'unknown',
     :                                    form='formatted')
#endif
#endif
#ifdef BINA
#ifdef SINB
      open  (unit = cnunit, file = 'system4.inp', status = 'unknown',
     :                                    form='formatted')
#endif
#endif

c     Title of run.
        read (cnunit, '(/A6)') run
c     Input file prefix.
        read (cnunit, '(/A6)') input

c     Integration stepsize.
read(cnunit,'(/F10.8)' )                hh(0)
c     Number of data points
read(cnunit, '(/I7)')               npoints
c     MC steps in predictor
c     **phase_a
read(cnunit, '(/I9)')               pstep_a
c     **phase_b
read(cnunit, '(/I9)')               pstep_b
c     MC steps in corrector
c     **phase_a
read(cnunit, '(/I9)')               cstep_a
c     **phase_b
read(cnunit, '(/I9)')               cstep_b
c     Iterations in corrector
c     ** phase_a
        read(cnunit, '(/I7)')                ncorr
c     MC steps in production runs
c     **phase_a
read(cnunit, '(/I9)')               fstep_a
c     **phase_b
```

```
read(cnunit, '(/I9)')               fstep_b

#ifdef SINA
c     Lennard-Jones parameters for the species.
c      **phase_a
read(cnunit, '(/F6.3)')         sigma_a
read(cnunit, '(/F6.3)')         eps_a
#endif
#ifdef BINA
c     Number of particles assigned to each species.
c      **phase_a
read(cnunit, '(/2I7)')                N_a

c     Lennard-Jones parameters for the species.
c      **phase_a
read(cnunit, '(/4F6.3)')         sigmas_a
read(cnunit, '(/4F6.3)')         eps_a

c     Henry's law constant for dilute component
c      **phase_a
read(cnunit, '(/E12.5)')           hen2_a
#endif
#ifdef SINB
c     Lennard-Jones parameters for the species.
c      **phase_b
read(cnunit, '(/F6.3)')         sigma_b
read(cnunit, '(/F6.3)')         eps_b
#endif
#ifdef BINB
c     Number of particles assigned to each species.
c      **phase_b
read(cnunit, '(/2I7)')                N_b

c     Lennard-Jones parameters for the species.
c      **phase_b
read(cnunit, '(/4F6.3)')         sigmas_b
read(cnunit, '(/4F6.3)')         eps_b

c     Henry's law constant for dilute component.
c      **phase_b
read(cnunit, '(/E12.5)')           hen2_b
#endif

c  Initial condition data.
c      Reduced temperature and pressure.
read(cnunit, '(/2F9.4)') TEMP,P
c      Enthalpy of each phase.
read(cnunit, '(/2F9.4)') h_a,h_b
c      Volume of each phase.
read(cnunit, '(/2F9.2)') vol_a,vol_b
c      Fugacity fraction of component 2.
        read(cnunit, '(/F10.8) ') fug_frac
c      Number of attempted interchanges
read(cnunit, '(/I7)') NSWAP
c      Limits on product hh(k-1)*F(k) for determining  hh(k)
read(cnunit, '(/2F9.4)') maxlimit,minlimit


      close (unit = cnunit)

      return

      end
```

```
       subroutine readcn_a ( cnfile )


C    ********************************************************************
C    ** SUBROUTINE TO READ IN THE CONFIGURATION FROM UNIT 10          **
C    ********************************************************************

        CHARACTER   cnfile*30
        REAL        RX(NOPA), RY(NOPA), RZ(NOPA)
#ifdef SSE
        REAL        vol_a,box,h_a,cell_con_a
#else
        REAL        vol_a,box,h_a
#endif
        REAL        rrbox


        INTEGER     CNUNIT, i
        PARAMETER ( CNUNIT = 10 )
        INTEGER     NN

        COMMON / config_a / RX, RY, RZ
#ifdef SSE
        common / block13 / vol_a,box,h_a,cell_con_a
#else
        common / block13 / vol_a,box,h_a
#endif
C   ********************************************************************

        OPEN ( UNIT = CNUNIT, FILE = cnfile, STATUS = 'OLD')
#ifdef SSE
        READ (CNUNIT,*) NN, box, cell_con_a
#else
        READ (CNUNIT,*) NN, box
#endif

c        READ (CNUNIT,*) RX, RY, RZ
         DO i=1,NOPA
         READ (CNUNIT,*) RX(i), RY(i), RZ(i)
         ENDDO

        CLOSE ( UNIT = CNUNIT )

        RETURN
        END
```

```
       SUBROUTINE writcn_a ( cnfile )


C    ********************************************************************
C    ** SUBROUTINE TO WRITE OUT THE CONFIGURATION TO UNIT 10          **
C    ********************************************************************


        CHARACTER   cnfile*30
        REAL        RX(NOPA), RY(NOPA), RZ(NOPA)
        REAL        vol_a,BOX,h_a
        integer update, simcount,i
#ifdef BINA
        integer     species(NOPA)
        integer     N_a(NCOMPA)
#endif
        INTEGER     CNUNIT
        PARAMETER ( CNUNIT = 10 )

        COMMON / config_a / RX, RY, RZ
#ifdef SSE
        common / block13 / vol_a,BOX,h_a,cell_con_a
#else
        common / block13 / vol_a,BOX,h_a
#endif

#ifdef BINA
        common / block2 / species,N_a
#endif
         common / block4 / update,simcount
C   ********************************************************************

        OPEN ( UNIT = CNUNIT, FILE = cnfile, STATUS = 'UNKNOWN')
c     :         FORM = 'UNFORMATTED'                        )

#ifdef SSE
        WRITE ( CNUNIT,* ) NOPA, BOX, cell_con_a
#else
        WRITE ( CNUNIT,* ) NOPA, BOX
#endif
c        WRITE ( CNUNIT,* ) RX, RY, RZ
         DO i=1,NOPA
         WRITE ( CNUNIT,* ) RX(i), RY(i), RZ(i)
         ENDDO

        CLOSE ( UNIT = CNUNIT )

c   OPEN ( UNIT = 9, FILE = "view_a.xyz", STATUS = 'UNKNOWN')
c write(9,*) NOPA
c write(9,*)
c#ifdef SINA
c do 10 i=1,NOPA
c write(9,*) 'C',rx(i),ry(i),rz(i)
c10     continue
c#endif
c#ifdef BINA

c       do  i=1,N_a(1)
c          write(9,*) 'C',rx(i),ry(i),rz(i)
c       enddo
c       do i=1+N_a(1),N_a(1)+N_a(2)
c          write(9,*) 'I',rx(i),ry(i),rz(i)
```

```
c       enddo
c#endif

c       close(unit=9)


        RETURN
        END



       SUBROUTINE writpos_a (posfile)


C    ********************************************************************
C    ** SUBROUTINE TO WRITE OUT THE CONFIGURATION TO UNIT 10          **
C    ********************************************************************


        CHARACTER   cnfile*30,run*6,posfile*30
        REAL        RX(NOPA), RY(NOPA), RZ(NOPA)
        REAL        vol_a,BOX,h_a
        integer update, simcount
#ifdef BINA
        integer     species(NOPA)
        integer     N_a(NCOMPA)
#endif
        INTEGER     CNUNIT


        COMMON / config_a / RX, RY, RZ
#ifdef SSE
        common / block13 / vol_a,BOX,h_a,cell_con_a
#else
        common / block13 / vol_a,BOX,h_a
#endif

#ifdef BINA
        common / block2 / species,N_a
#endif
         common / block4 / update,simcount
         common / block6 / infile,outfile,run,cnfile
C   ********************************************************************


          CNUNIT =10
  OPEN ( UNIT = CNUNIT, FILE = posfile,
     :              STATUS = 'UNKNOWN')
c   OPEN ( UNIT = CNUNIT, FILE = run//'_apos.xyz',
c    :              STATUS = 'UNKNOWN')



write(CNUNIT,*) NOPA
write(CNUNIT,*)
#ifdef SINA
do 10 i=1,NOPA
write(CNUNIT,*) 'C',rx(i),ry(i),rz(i)
10     continue
#endif
#ifdef BINA

       do  i=1,N_a(1)
```

```
          write(CNUNIT,*) 'C',rx(i),ry(i),rz(i)
        enddo
        do i=1+N_a(1),N_a(1)+N_a(2)
          write(CNUNIT,*) 'I',rx(i),ry(i),rz(i)
        enddo
#endif
         CLOSE (UNIT = CNUNIT)

        RETURN
        END


        subroutine sumup_a ( RCUT, RMIN, OVRLAP, V12, V6, W12, W6 )


C    ********************************************************************
C    ** CALCULATES THE TOTAL POTENTIAL ENERGY FOR A CONFIGURATION.     **
C    **                                                                **
C    ** PRINCIPAL VARIABLES:                                           **
C    **                                                                **
C    ** INTEGER N                 THE NUMBER OF ATOMS                  **
C    ** REAL    RX(N(,RY(N),RZ(N) THE POSITIONS OF THE ATOMS           **
C    ** REAL    V                 THE POTENTIAL ENERGY                 **
C    ** REAL    W                 THE VIRIAL                           **
C    ** LOGICAL OVRLAP            TRUE FOR SUBSTANTIAL ATOM OVERLAP    **
C    **                                                                **
C    ** USAGE:                                                         **
C    **                                                                **
C    ** THE SUBROUTINE RETURNS THE TOTAL POTENTIAL ENERGY AT THE       **
C    ** BEGINNING AND END OF THE RUN.                                  **
C    ********************************************************************


        REAL        RX(NOPA), RY(NOPA), RZ(NOPA)
        REAL        RMIN, RCUT, V12, V6, W12, W6
#ifdef SSE
        REAL        avVOL,BOX,avh, cell_con_a
#else
        REAL        avVOL,BOX,avh
#endif
        LOGICAL     OVRLAP

        REAL        RCUTSQ, RMINSQ, VIJ12, VIJ6
        REAL        RXI, RYI, RZI, RXIJ, RYIJ, RZIJ
        REAL        SR2, SR6, RIJSQ, BOXINV
#ifdef SINA
        real      sigma,eps
#endif
#ifdef BINA
        integer species(NOPA),N_a(NCOMPA)
        real    sig2_a(NCOMPA,NCOMPA)
        real    sig3_a(NCOMPA,NCOMPA)
        real    eps_a4(NCOMPA,NCOMPA)
#endif
        INTEGER     I, J

        COMMON / config_a / RX, RY, RZ
#ifdef SSE
        common / block13 / avVOL,BOX,avh,cell_con_a
#else
        common / block13 / avVOL,BOX,avh
```

```
#endif
#ifdef SINA
        common / sina1 / sigma,eps
#endif
#ifdef BINA
         common / bina2 / sig2_a,sig3_a,eps_a4
         common / block2 / species,N_a
#endif

C    ******************************************************************

        OVRLAP = .FALSE.
        RCUTSQ = RCUT * RCUT
        RMINSQ = RMIN * RMIN
        BOXINV = 1.0 / BOX

        V12    = 0.0
        V6     = 0.0
        W12    = 0.0
        W6     = 0.0

C    ** LOOP OVER ALL THE PAIRS IN THE LIQUID **

        DO 100 I = 1, NOPA - 1

           RXI = RX(I)
           RYI = RY(I)
           RZI = RZ(I)

           DO 99 J = I + 1, NOPA

              RXIJ  = RXI - RX(J)
              RYIJ  = RYI - RY(J)
              RZIJ  = RZI - RZ(J)
c             **  minimum image **
              RXIJ  = RXIJ - ANINT ( RXIJ * BOXINV ) * BOX
              RYIJ  = RYIJ - ANINT ( RYIJ * BOXINV ) * BOX
              RZIJ  = RZIJ - ANINT ( RZIJ * BOXINV ) * BOX

              RIJSQ = RXIJ * RXIJ + RYIJ * RYIJ + RZIJ * RZIJ

              IF ( RIJSQ .LT. RMINSQ ) THEN

                 OVRLAP = .TRUE.
                 RETURN

              ELSEIF ( RIJSQ .LT. RCUTSQ ) THEN
#ifdef SINA
                 SR2   = (sigma*sigma) / RIJSQ
                 SR6   = SR2 * SR2 * SR2
                 VIJ12 = SR6 *  SR6
                 VIJ6  = - SR6
#endif
#ifdef BINA
                 SR2   = sig2_a(species(i),species(j)) / RIJSQ
                 SR6   = SR2 * SR2 * SR2
                 VIJ12 = SR6 * SR6 * eps_a4(species(i),species(j))
                 VIJ6  = - SR6 * eps_a4(species(i),species(j))
#endif


                 V12   = V12 + VIJ12
                 V6    = V6  + VIJ6
```

```
                W12   = W12 + VIJ12
                W6    = W6  + VIJ6 * 0.5

             ENDIF

99         CONTINUE

100     CONTINUE

#ifdef SINA
        V12 = 4.0 *eps* V12
        V6  = 4.0 *eps* V6
        W12 = 48.0 *eps* W12 / 3.0
        W6  = 48.0 *eps* W6  / 3.0
#endif
#ifdef BINA
        W12 = 12.0 * W12 / 3.0
        W6  = 12.0 * W6  / 3.0
#endif
        RETURN
        END



        subroutine energy_a ( RXI, RYI, RZI, I, RCUT,
     :                       V12, V6, W12, W6 )


C    ***********************************************************************
C    ** CALCULATES THE POTENTIAL ENERGY OF I WITH ALL OTHER ATOMS     **
C    **                                                              **
C    ** PRINCIPAL VARIABLES:                                         **
C    **                                                              **
C    ** INTEGER I                THE ATOM OF INTEREST                **
C    ** INTEGER N                THE NUMBER OF ATOMS                 **
C    ** REAL    RX(N),RY(N),RZ(N) THE ATOM POSITIONS                 **
C    ** REAL    RXI,RYI,RZI      THE COORDINATES OF ATOM I           **
C    ** REAL    V                THE POTENTIAL ENERGY OF ATOM I      **
C    ** REAL    W                THE VIRIAL OF ATOM I                **
C    **                                                              **
C    ** USAGE:                                                       **
C    **                                                              **
C    ** THIS SUBROUTINE IS USED TO CALCULATE THE CHANGE OF ENERGY     **
C    ** DURING A TRIAL MOVE OF ATOM I. IT IS CALLED BEFORE AND       **
C    ** AFTER THE RANDOM DISPLACEMENT OF I.                          **
C    ***********************************************************************
        REAL        RX(NOPA), RY(NOPA), RZ(NOPA)
        REAL        RCUT, V12, V6, W12, W6
#ifdef SSE
        REAL        avVOL,BOX,avh, cell_con_a
#else
        REAL        avVOL,BOX,avh
#endif

        REAL        RCUTSQ, VIJ12, VIJ6
        REAL        RXI, RYI, RZI, RXIJ, RYIJ, RZIJ
        REAL        SR2, SR6, RIJSQ, BOXINV
#ifdef SINA
        real        sigma,eps
        real        sigsq
#endif
#ifdef BINA
        integer     species(NOPA),N_a(NCOMPA)
```

```
        real        sig2_a(NCOMPA,NCOMPA),sig3_a(NCOMPA,NCOMPA)
        real        eps_a4(NCOMPA,NCOMPA)
#endif
        INTEGER     I, J

        COMMON / config_a / RX, RY, RZ
#ifdef SSE
        common / block13 / avVOL,BOX,avh,cell_con_a
#else
        common / block13 / avVOL,BOX,avh
#endif
#ifdef SINA
        common / sina1 / sigma,eps
#endif
#ifdef BINA
         common / bina2 / sig2_a,sig3_a,eps_a4
         common / block2 / species,N_a
#endif


C     *****************************************************************
#ifdef SINA
         sigsq  = sigma* sigma
#endif

        RCUTSQ = RCUT * RCUT
        BOXINV = 1.0 / BOX

        V12    = 0.0
        V6     = 0.0
        W12    = 0.0
        W6     = 0.0

C    ** LOOP OVER ALL MOLECULES EXCEPT I  **

        DO 100 J = 1, NOPA

        IF ( I .NE. J ) THEN

           RXIJ  = RXI - RX(J)
           RYIJ  = RYI - RY(J)
           RZIJ  = RZI - RZ(J)
c          ** minimum image **
           RXIJ  = RXIJ - ANINT ( RXIJ * BOXINV ) * BOX
           RYIJ  = RYIJ - ANINT ( RYIJ * BOXINV ) * BOX
           RZIJ  = RZIJ - ANINT ( RZIJ * BOXINV ) * BOX

           RIJSQ = RXIJ * RXIJ + RYIJ * RYIJ + RZIJ * RZIJ

           IF ( RIJSQ .LT. RCUTSQ ) THEN

#ifdef SINA
              SR2   = sigsq/ RIJSQ
              SR6   = SR2 * SR2 * SR2
              VIJ12 = SR6 * SR6
              VIJ6  = - SR6
#endif
#ifdef BINA
              SR2   = sig2_a(species(i),species(j)) / RIJSQ
              SR6   = SR2 * SR2 * SR2
              VIJ12 = SR6 * SR6 * eps_a4(species(i),species(j))
              VIJ6  = - SR6 * eps_a4(species(i),species(j))
#endif
```

```
               V12   = V12 + VIJ12
               V6    = V6  + VIJ6
               W12   = W12 + VIJ12
               W6    = W6  + VIJ6 * 0.5
             ENDIF


           ENDIF

100        CONTINUE

#ifdef SINA
           V12 = 4.0 *eps* V12
           V6  = 4.0 *eps* V6
           W12 = 48.0 *eps* W12 / 3.0
           W6  = 48.0 *eps* W6  / 3.0
#endif
#ifdef BINA
           W12 = 12.0 * W12 / 3.0
           W6  = 12.0 * W6  / 3.0
#endif

           RETURN
           END



           subroutine energi_a ( RXI, RYI, RZI, I, RCUT,
     :                           V12, V6, W12, W6,tryiden )


C    *****************************************************************
C    ** CALCULATES THE POTENTIAL ENERGY OF I WITH ALL OTHER ATOMS     **
C    **                                                              **
C    ** PRINCIPAL VARIABLES:                                         **
C    **                                                              **
C    ** INTEGER I                THE ATOM OF INTEREST                **
C    ** INTEGER N                THE NUMBER OF ATOMS                 **
C    ** REAL    RX(N),RY(N),RZ(N) THE ATOM POSITIONS                 **
C    ** REAL    RXI,RYI,RZI      THE COORDINATES OF ATOM I           **
C    ** REAL    V                THE POTENTIAL ENERGY OF ATOM I      **
C    ** REAL    W                THE VIRIAL OF ATOM I                **
C    **                                                              **
C    ** USAGE:                                                       **
C    **                                                              **
C    ** THIS SUBROUTINE IS USED TO CALCULATE THE CHANGE OF ENERGY     **
C    ** DURING A TRIAL MOVE OF ATOM I. IT IS CALLED BEFORE AND        **
C    ** AFTER THE RANDOM DISPLACEMENT OF I.                          **
C    *****************************************************************
           REAL        RX(NOPA), RY(NOPA), RZ(NOPA)
           REAL        RCUT, V12, V6, W12, W6
#ifdef SSE
           REAL        avVOL,BOX,avh,cell_con_a
#else
           REAL        avVOL,BOX,avh
#endif

           REAL        RCUTSQ, VIJ12, VIJ6
           REAL        RXI, RYI, RZI, RXIJ, RYIJ, RZIJ
           REAL        SR2, SR6, RIJSQ, BOXINV

#ifdef BINA
```

```
        integer     species(NOPA),N_a(NCOMPA)
        real        sig2_a(NCOMPA,NCOMPA),sig3_a(NCOMPA,NCOMPA)
        real        eps_a4(NCOMPA,NCOMPA)
        integer     tryiden
#endif
        INTEGER     I, J

        COMMON / config_a / RX, RY, RZ
#ifdef SSE
        common / block13 / avVOL,BOX,avh,cell_con_a
#else
        common / block13 / avVOL,BOX,avh
#endif

#ifdef BINA
         common / bina2 / sig2_a,sig3_a,eps_a4
         common / block2 / species,N_a
#endif


C     ******************************************************************

        RCUTSQ = RCUT * RCUT
        BOXINV = 1.0 / BOX

        V12    = 0.0
        V6     = 0.0
        W12    = 0.0
        W6     = 0.0

C    ** LOOP OVER ALL MOLECULES EXCEPT I  **

        DO 100 J = 1, NOPA

        IF ( I .NE. J ) THEN

           RXIJ  = RXI - RX(J)
           RYIJ  = RYI - RY(J)
           RZIJ  = RZI - RZ(J)
c          ** minimum image **
           RXIJ  = RXIJ - ANINT ( RXIJ * BOXINV ) * BOX
           RYIJ  = RYIJ - ANINT ( RYIJ * BOXINV ) * BOX
           RZIJ  = RZIJ - ANINT ( RZIJ * BOXINV ) * BOX

           RIJSQ = RXIJ * RXIJ + RYIJ * RYIJ + RZIJ * RZIJ

           IF ( RIJSQ .LT. RCUTSQ ) THEN


c#ifdef BINA
              SR2   = sig2_a(tryiden,species(j)) / RIJSQ
              SR6   = SR2 * SR2 * SR2
              VIJ12 = SR6 * SR6 * eps_a4(tryiden,species(j))
              VIJ6  = - SR6 * eps_a4(tryiden,species(j))
c#endif

              V12   = V12 + VIJ12
              V6    = V6  + VIJ6
              W12   = W12 + VIJ12
              W6    = W6  + VIJ6 * 0.5
           ENDIF
```

```
        ENDIF

100     CONTINUE


c#ifdef BINA
        W12 = 12.0 * W12 / 3.0
        W6  = 12.0 * W6  / 3.0
c#endif

        RETURN
        END


       subroutine readcn_b ( cnfile )


C    ***********************************************************************
C    ** SUBROUTINE TO READ IN THE CONFIGURATION FROM UNIT 10          **
C    ***********************************************************************

        CHARACTER   cnfile*30
        REAL        RX(NOPB), RY(NOPB), RZ(NOPB)
#ifdef SLE
        REAL        vol_b,box,h_b,cell_con
#else
        REAL        vol_b,box,h_b
#endif

        INTEGER     CNUNIT, i
        PARAMETER ( CNUNIT = 10 )
        INTEGER     NN

        COMMON / config_b / RX, RY, RZ
#ifdef SLE
        common / block14 / vol_b,box,h_b,cell_con
#else
        common / block14 / vol_b,box,h_b
#endif
C   ***********************************************************************

        OPEN ( UNIT = CNUNIT, FILE = cnfile, STATUS = 'OLD')
#ifdef SLE
        READ (CNUNIT,*) NN, box, cell_con
#else
        READ (CNUNIT,*) NN, box
#endif
c       READ (CNUNIT,*) RX, RY, RZ
         DO i = 1,NOPB
         READ (CNUNIT,*) RX(i), RY(i), RZ(i)
         ENDDO

        CLOSE ( UNIT = CNUNIT )

        RETURN
        END


       SUBROUTINE writcn_b ( cnfile )


C    ***********************************************************************
```

```
C    ** SUBROUTINE TO WRITE OUT THE CONFIGURATION TO UNIT 10          **
C    ***********************************************************************


        CHARACTER   cnfile*30
        REAL        RX(NOPB), RY(NOPB), RZ(NOPB)
        REAL        vol_b,BOX,h_b,cell_con
        integer     update, simcount,i
#ifdef BINB
        integer     species(NOPB)
        integer     N_b(NCOMPB)
#endif
        INTEGER     CNUNIT
        PARAMETER ( CNUNIT = 10 )

        COMMON / config_b / RX, RY, RZ

#ifdef SLE
        common / block14 / vol_b,BOX,h_b,cell_con
#else
        common / block14 / vol_b,BOX,h_b
#endif
#ifdef BINB
        common / block12 / species,N_b
#endif
         common / block4 / update,simcount

C   ***********************************************************************

        OPEN ( UNIT = CNUNIT, FILE = cnfile, STATUS = 'UNKNOWN')
c     :         FORM = 'UNFORMATTED'                        )

#ifdef SLE
        WRITE ( CNUNIT,* ) NOPB, BOX, cell_con
#else
        WRITE ( CNUNIT,* ) NOPB, BOX
#endif
c       WRITE ( CNUNIT,* ) RX, RY, RZ
         DO i=1,NOPB
         WRITE ( CNUNIT,* ) RX(i), RY(i), RZ(i)
         ENDDO

        CLOSE ( UNIT = CNUNIT )

c   OPEN ( UNIT = 9, FILE = "view_b.xyz", STATUS = 'UNKNOWN')
c write(9,*) NOPB
c write(9,*)
c#ifdef SINB
c do 10 i=1,NOPB
c write(9,*) 'C',rx(i),ry(i),rz(i)
c10     continue
c#endif
c#ifdef BINB

c       do  i=1,N_b(1)
c          write(9,*) 'C',rx(i),ry(i),rz(i)
c       enddo
c       do i=1+N_b(1),N_b(1)+N_b(2)
c          write(9,*) 'I',rx(i),ry(i),rz(i)
c       enddo
c#endif

c       close(unit=9)
```

```
        RETURN
        END


       SUBROUTINE writpos_b (posfile )

C    ********************************************************************
C    ** SUBROUTINE TO WRITE OUT THE CONFIGURATION TO UNIT 10          **
C    ********************************************************************


        CHARACTER   cnfile*30,run*6,posfile*30
        REAL        RX(NOPB), RY(NOPB), RZ(NOPB)
        REAL        vol_b,BOX,h_b,cell_con
        integer     update, simcount
#ifdef BINB
        integer     species(NOPB)
        integer     N_b(NCOMPB)
#endif
        INTEGER     CNUNIT

        COMMON / config_b / RX, RY, RZ

#ifdef SLE
        common / block14 / vol_b,BOX,h_b,cell_con
#else
        common / block14 / vol_b,BOX,h_b
#endif
#ifdef BINB
        common / block12 / species,N_b
#endif
         common / block4 / update,simcount
         common / block6 / infile,outfile,run,cnfile
C   ********************************************************************

         CNUNIT =10

  OPEN ( UNIT = CNUNIT, FILE = posfile,
     :              STATUS = 'UNKNOWN')
c   OPEN ( UNIT = CNUNIT, FILE = run//'_bpos.xyz',
c    :              STATUS = 'UNKNOWN')


write(CNUNIT,*) NOPB
write(CNUNIT,*)
#ifdef SINB
do 10 i=1,NOPB
write(CNUNIT,*) 'C',rx(i),ry(i),rz(i)
10     continue
#endif
#ifdef BINB

       do  i=1,N_b(1)
          write(CNUNIT,*) 'C',rx(i),ry(i),rz(i)
       enddo
       do i=1+N_b(1),N_b(1)+N_b(2)
          write(CNUNIT,*) 'I',rx(i),ry(i),rz(i)
       enddo
#endif
```

```
       CLOSE(UNIT = CNUNIT)

        RETURN
        END

           subroutine sumup_b ( RCUT, RMIN, OVRLAP, V12, V6, W12, W6 )


C     ***********************************************************************
C     ** CALCULATES THE TOTAL POTENTIAL ENERGY FOR A CONFIGURATION.     **
C     **                                                                   **
C     ** PRINCIPAL VARIABLES:                                            **
C     **                                                                   **
C     ** INTEGER N                 THE NUMBER OF ATOMS                   **
C     ** REAL    RX(N(,RY(N),RZ(N) THE POSITIONS OF THE ATOMS            **
C     ** REAL    V                 THE POTENTIAL ENERGY                  **
C     ** REAL    W                 THE VIRIAL                            **
C     ** LOGICAL OVRLAP            TRUE FOR SUBSTANTIAL ATOM OVERLAP     **
C     **                                                                   **
C     ** USAGE:                                                          **
C     **                                                                   **
C     ** THE SUBROUTINE RETURNS THE TOTAL POTENTIAL ENERGY AT THE        **
C     ** BEGINNING AND END OF THE RUN.                                   **
C     ***********************************************************************

        REAL        RX(NOPB), RY(NOPB), RZ(NOPB)
        REAL        RMIN, RCUT, V12, V6, W12, W6
#ifdef SLE
        REAL        avVOL,BOX,avh, cell_con
#else
        REAL        avVOL,BOX,avh
#endif
        LOGICAL     OVRLAP

        REAL        RCUTSQ, RMINSQ, VIJ12, VIJ6
        REAL        RXI, RYI, RZI, RXIJ, RYIJ, RZIJ
        REAL        SR2, SR6, RIJSQ, BOXINV
#ifdef SINB
        real      sigma,eps
#endif
#ifdef BINB
        integer species(NOPB),N_b(NCOMPB)
        real    sig2_b(NCOMPB,NCOMPB),sig3_b(NCOMPB,NCOMPB)
        real    eps_b4(NCOMPB,NCOMPB)
#endif
        INTEGER     I, J

        COMMON / config_b / RX, RY, RZ
#ifdef SLE
        common / block14 / avVOL,BOX,avh,cell_con
#else
        common / block14 / avVOL,BOX,avh
#endif

#ifdef SINB
        common / sinb1 / sigma,eps
#endif
#ifdef BINB
         common / binb2 / sig2_b,sig3_b,eps_b4
         common / block12 / species,N_b
```

```
#endif

C     ***********************************************************************

         OVRLAP = .FALSE.
         RCUTSQ = RCUT * RCUT
         RMINSQ = RMIN * RMIN
         BOXINV = 1.0 / BOX

         V12    = 0.0
         V6     = 0.0
         W12    = 0.0
         W6     = 0.0

C     ** LOOP OVER ALL THE PAIRS IN THE LIQUID **

         DO 100 I = 1, NOPB - 1

            RXI = RX(I)
            RYI = RY(I)
            RZI = RZ(I)

            DO 99 J = I + 1, NOPB

               RXIJ  = RXI - RX(J)
               RYIJ  = RYI - RY(J)
               RZIJ  = RZI - RZ(J)
c              **  minimum image **
               RXIJ  = RXIJ - ANINT ( RXIJ * BOXINV ) * BOX
               RYIJ  = RYIJ - ANINT ( RYIJ * BOXINV ) * BOX
               RZIJ  = RZIJ - ANINT ( RZIJ * BOXINV ) * BOX

               RIJSQ = RXIJ * RXIJ + RYIJ * RYIJ + RZIJ * RZIJ

               IF ( RIJSQ .LT. RMINSQ ) THEN

                  OVRLAP = .TRUE.
                  RETURN

               ELSEIF ( RIJSQ .LT. RCUTSQ ) THEN
#ifdef SINB
                  SR2   = (sigma*sigma) / RIJSQ
                  SR6   = SR2 * SR2 * SR2
                  VIJ12 = SR6 *  SR6
                  VIJ6  = - SR6
#endif
#ifdef BINB
                  SR2   = sig2_b(species(i),species(j)) / RIJSQ
                  SR6   = SR2 * SR2 * SR2
                  VIJ12 = SR6 * SR6 * eps_b4(species(i),species(j))
                  VIJ6  = - SR6 * eps_b4(species(i),species(j))
#endif
                  V12   = V12 + VIJ12
                  V6    = V6  + VIJ6
                  W12   = W12 + VIJ12
                  W6    = W6  + VIJ6 * 0.5

               ENDIF

99          CONTINUE

100      CONTINUE
```

```
#ifdef SINB
        V12 = 4.0 *eps* V12
        V6  = 4.0 *eps* V6
        W12 = 48.0 *eps* W12 / 3.0
        W6  = 48.0 *eps* W6  / 3.0
#endif
#ifdef BINB
        W12 = 12.0 * W12 / 3.0
        W6  = 12.0 * W6  / 3.0
#endif
        RETURN
        END


        subroutine energy_b ( RXI, RYI, RZI, I, RCUT,
     :                       V12, V6, W12, W6 )


C    ***********************************************************************
C    ** CALCULATES THE POTENTIAL ENERGY OF I WITH ALL OTHER ATOMS     **
C    **                                                               **
C    ** PRINCIPAL VARIABLES:                                          **
C    **                                                               **
C    ** INTEGER I                THE ATOM OF INTEREST                 **
C    ** INTEGER N                THE NUMBER OF ATOMS                  **
C    ** REAL    RX(N),RY(N),RZ(N) THE ATOM POSITIONS                  **
C    ** REAL    RXI,RYI,RZI      THE COORDINATES OF ATOM I            **
C    ** REAL    V                THE POTENTIAL ENERGY OF ATOM I       **
C    ** REAL    W                THE VIRIAL OF ATOM I                 **
C    **                                                               **
C    ** USAGE:                                                        **
C    **                                                               **
C    ** THIS SUBROUTINE IS USED TO CALCULATE THE CHANGE OF ENERGY     **
C    ** DURING A TRIAL MOVE OF ATOM I. IT IS CALLED BEFORE AND        **
C    ** AFTER THE RANDOM DISPLACEMENT OF I.                           **
C    ***********************************************************************
        REAL        RX(NOPB), RY(NOPB), RZ(NOPB)
        REAL        RCUT, V12, V6, W12, W6
#ifdef SLE
        REAL        avVOL,BOX,avh,cell_con
#else
        REAL        avVOL,BOX,avh
#endif

        REAL        RCUTSQ, VIJ12, VIJ6
        REAL        RXI, RYI, RZI, RXIJ, RYIJ, RZIJ
        REAL        SR2, SR6, RIJSQ, BOXINV
#ifdef SINB
        real        sigma,eps
        real        sigsq
#endif
#ifdef BINB
        integer     species(NOPB),N_b(NCOMPB)
        real        sig2_b(NCOMPB,NCOMPB),sig3_b(NCOMPB,NCOMPB)
        real        eps_b4(NCOMPB,NCOMPB)
#endif
        INTEGER     I, J

        COMMON / config_b / RX, RY, RZ

#ifdef SLE
        common / block14 / avVOL,BOX,avh,cell_con
#else
```

```
        common / block14 / avVOL,BOX,avh
#endif
#ifdef SINB
        common / sinb1 / sigma,eps
#endif
#ifdef BINB
         common / binb2 / sig2_b,sig3_b,eps_b4
         common / block12 / species,N_b
#endif


C     ******************************************************************
#ifdef SINB
         sigsq  = sigma* sigma
#endif

        RCUTSQ = RCUT * RCUT
        BOXINV = 1.0 / BOX

        V12    = 0.0
        V6     = 0.0
        W12    = 0.0
        W6     = 0.0

C    ** LOOP OVER ALL MOLECULES EXCEPT I  **

        DO 100 J = 1, NOPB

        IF ( I .NE. J ) THEN

           RXIJ  = RXI - RX(J)
           RYIJ  = RYI - RY(J)
           RZIJ  = RZI - RZ(J)
c          ** minimum image **
           RXIJ  = RXIJ - ANINT ( RXIJ * BOXINV ) * BOX
           RYIJ  = RYIJ - ANINT ( RYIJ * BOXINV ) * BOX
           RZIJ  = RZIJ - ANINT ( RZIJ * BOXINV ) * BOX

           RIJSQ = RXIJ * RXIJ + RYIJ * RYIJ + RZIJ * RZIJ

           IF ( RIJSQ .LT. RCUTSQ ) THEN

#ifdef SINB
              SR2   = sigsq/ RIJSQ
              SR6   = SR2 * SR2 * SR2
              VIJ12 = SR6 * SR6
              VIJ6  = - SR6
#endif
#ifdef BINB
              SR2   = sig2_b(species(i),species(j)) / RIJSQ
              SR6   = SR2 * SR2 * SR2
              VIJ12 = SR6 * SR6 * eps_b4(species(i),species(j))
              VIJ6  = - SR6 * eps_b4(species(i),species(j))
#endif

              V12   = V12 + VIJ12
              V6    = V6  + VIJ6
              W12   = W12 + VIJ12
              W6    = W6  + VIJ6 * 0.5
           ENDIF


        ENDIF
```

```
100     CONTINUE

#ifdef SINB
        V12 = 4.0 *eps* V12
        V6  = 4.0 *eps* V6
        W12 = 48.0 *eps* W12 / 3.0
        W6  = 48.0 *eps* W6  / 3.0
#endif
#ifdef BINB
        W12 = 12.0 * W12 / 3.0
        W6  = 12.0 * W6  / 3.0
#endif

        RETURN
        END



        subroutine energi_b ( RXI, RYI, RZI, I, RCUT,
     :                      V12, V6, W12, W6,tryiden )


C    *****************************************************************
C    ** CALCULATES THE POTENTIAL ENERGY OF I WITH ALL OTHER ATOMS     **
C    **                                                              **
C    ** PRINCIPAL VARIABLES:                                         **
C    **                                                              **
C    ** INTEGER I                THE ATOM OF INTEREST                **
C    ** INTEGER N                THE NUMBER OF ATOMS                 **
C    ** REAL    RX(N),RY(N),RZ(N) THE ATOM POSITIONS                 **
C    ** REAL    RXI,RYI,RZI      THE COORDINATES OF ATOM I           **
C    ** REAL    V                THE POTENTIAL ENERGY OF ATOM I      **
C    ** REAL    W                THE VIRIAL OF ATOM I                **
C    **                                                              **
C    ** USAGE:                                                       **
C    **                                                              **
C    ** THIS SUBROUTINE IS USED TO CALCULATE THE CHANGE OF ENERGY     **
C    ** DURING A TRIAL MOVE OF ATOM I. IT IS CALLED BEFORE AND       **
C    ** AFTER THE RANDOM DISPLACEMENT OF I.                          **
C    *****************************************************************
        REAL        RX(NOPB), RY(NOPB), RZ(NOPB)
        REAL        RCUT, V12, V6, W12, W6
#ifdef SLE
        REAL        avVOL,BOX,avh,cell_con
#else
        REAL        avVOL,BOX,avh
#endif
        REAL        RCUTSQ, VIJ12, VIJ6
        REAL        RXI, RYI, RZI, RXIJ, RYIJ, RZIJ
        REAL        SR2, SR6, RIJSQ, BOXINV

#ifdef BINB
        integer     species(NOPB),N_b(NCOMPB)
        real        sig2_b(NCOMPB,NCOMPB),sig3_b(NCOMPB,NCOMPB)
        real        eps_b4(NCOMPB,NCOMPB)
        integer     tryiden
#endif
        INTEGER     I, J

        COMMON / config_b / RX, RY, RZ

#ifdef SLE
```

```
        common / block14 / avVOL,BOX,avh,cell_con
#else
        common / block14 / avVOL,BOX,avh
#endif

#ifdef BINB
         common / binb2 / sig2_b,sig3_b,eps_b4
         common / block12 / species,N_b
#endif


C     ******************************************************************

        RCUTSQ = RCUT * RCUT
        BOXINV = 1.0 / BOX

        V12    = 0.0
        V6     = 0.0
        W12    = 0.0
        W6     = 0.0

C    ** LOOP OVER ALL MOLECULES EXCEPT I  **

        DO 100 J = 1, NOPB

        IF ( I .NE. J ) THEN

           RXIJ  = RXI - RX(J)
           RYIJ  = RYI - RY(J)
           RZIJ  = RZI - RZ(J)
c          ** minimum image **
           RXIJ  = RXIJ - ANINT ( RXIJ * BOXINV ) * BOX
           RYIJ  = RYIJ - ANINT ( RYIJ * BOXINV ) * BOX
           RZIJ  = RZIJ - ANINT ( RZIJ * BOXINV ) * BOX

           RIJSQ = RXIJ * RXIJ + RYIJ * RYIJ + RZIJ * RZIJ

           IF ( RIJSQ .LT. RCUTSQ ) THEN


c#ifdef BINB
              SR2   = sig2_b(tryiden,species(j)) / RIJSQ
              SR6   = SR2 * SR2 * SR2
              VIJ12 = SR6 * SR6 * eps_b4(tryiden,species(j))
              VIJ6  = - SR6 * eps_b4(tryiden,species(j))
c#endif

              V12   = V12 + VIJ12
              V6    = V6  + VIJ6
              W12   = W12 + VIJ12
              W6    = W6  + VIJ6 * 0.5
           ENDIF


        ENDIF

100     CONTINUE


c#ifdef BINB
        W12 = 12.0 * W12 / 3.0
        W6  = 12.0 * W6  / 3.0
c#endif
```

```
        RETURN
        END


      FUNCTION INTEGRAND(DEPEND,H_A,H_B,X_A,X_B)

      REAL INTEGRAND,DEPEND,H_A,H_B,X_A,X_B

        INTEGRAND = (1.-EXP(DEPEND))*(H_A-H_B)/(X_A - X_B)

      RETURN

      END

c Main header file for the GDI_SLE_P program.

      integer i,j
      integer simcount,update,npoints
      integer pstep_a,cstep_a,ncorr,fstep_a
      integer pstep_b,cstep_b,fstep_b
      integer NSWAP, SFLAG
      integer tens,ones
      real maxlimit,minlimit

      real r,A,B,Cminus1,Cnaught,Cplus1,D,product

#ifdef SINA
      real    eps_a,sigma_a
#endif
#ifdef SINB
      real    eps_b,sigma_b
#endif
#ifdef BINA
      integer species_a(NOPA),N_a(NCOMPA)
      real    hen2_a
      real    x_a(NCOMPA),avx_a(NCOMPA)
      real    sigmas_a(NCOMPA,NCOMPA),eps_a(NCOMPA,NCOMPA)
      real    sig2_a(NCOMPA,NCOMPA),sig3_a(NCOMPA,NCOMPA)
      real    eps_a4(NCOMPA,NCOMPA)
      real    stdx2_a
#endif
#ifdef BINB
      integer species_b(NOPB),N_b(NCOMPB)
      real    hen2_b
      real    x_b(NCOMPB),avx_b(NCOMPB)
      real    sigmas_b(NCOMPB,NCOMPB),eps_b(NCOMPB,NCOMPB)
      real    sig2_b(NCOMPB,NCOMPB),sig3_b(NCOMPB,NCOMPB)
      real    eps_b4(NCOMPB,NCOMPB)
      real    stdx2_b
#endif
      real    P,TEMP,h_a,h_b,vol_a,vol_b,cell_con,fug_frac
      real    box_a,box_b

      real    depend(0:50),indpnd(0:50),F(0:50), hh(0:50)

      real    INTEGRAND

      character cnfile*30
      character infile*30
      character outfile*30
      character bfile*30
```

```
      character input*6
      character run*6
      character   xfilea*30,pfilea*30,dfilea*30,posfilea*30
      character   xfileb*30,pfileb*30,dfileb*30,posfileb*30

      common / block4 / update,simcount

      common / block6 / infile,outfile,run,cnfile

      common / block11 / pstep_a,cstep_a,ncorr,fstep_a
      common / block11 / pstep_b,cstep_b,fstep_b

      common / syspar / TEMP,P
      common / fugacity / fug_frac
      common / swap / NSWAP

#ifdef SSE
      common / block13 / vol_a,box_a,h_a,cell_con_a
#else
      common / block13 / vol_a,box_a,h_a
#endif

#ifdef SLE
       common / block14 / vol_b,box_b,h_b,cell_con
#else
       common / block14 / vol_b,box_b,h_b
#endif

#ifdef BINA
      common / bina1 / sigmas_a,eps_a
      common / bina2 / sig2_a,sig3_a,eps_a4
      common / block2 / species_a,N_a
      common / block9 / x_a,avx_a,stdx2_a
      common / block10 /xfilea,pfilea,dfilea

#endif
#ifdef BINB
      common / binb1 / sigmas_b,eps_b
      common / binb2 / sig2_b,sig3_b,eps_b4
      common / block12 / species_b,N_b
      common / block19 / x_b,avx_b,stdx2_b
       common / block20 /xfileb,pfileb,dfileb

#endif
#ifdef SINA
      common / sina1 / sigma_a,eps_a
#endif
#ifdef SINB
      common / sinb1 / sigma_b,eps_b
#endif

C*** Define these at the beginning  (the names can be different except DTIME)

REAL TARRAY(2),DELTA,EXTIME,DTIME

EXTERNAL DTIME
```