# EQUATION OF STATE PHASE EQUILIBRIA CALCULATIONS

APPROVED:

Gary A. Pope

K. Sepehrnoori

# EQUATION OF STATE PHASE
# EQUILIBRIA CALCULATIONS

BY

DOUGLAS ROGER PERSCHKE, B.S. CHEM.

THESIS

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN ENGINEERING

THE UNIVERSITY OF TEXAS AT AUSTIN

December, 1984

# ACKNOWLEDGMENTS

I gratefully acknowledge my supervising professors, Dr. Gary Pope and Dr. Kamy Sepehrnoori, for their assistance and encouragement.

I would also like to thank the Baker Foundation and the Phillips Petroleum Corporation for the fellowship support that they provided, and I also gratefully acknowledge the financial support of this project by the industrial supporters of the Center for Enhanced Oil and Gas Recovery: ADREF, Amoco, Arco, British Petroleum, Chevron, Cities Service, Getty, Marathon, Mobil, Phillips, Schlumberger, Shell, Sohio, Sun, Tenneco, Union, W. R. Grace.

I would like to acknowledge Kent Thele for the many discussions which contributed to this thesis. Finally, I gratefully acknowledge John Clifford for the excellent job he did in typing this thesis.

Douglas R. Perschke
The University of Texas at Austin
Austin, Texas

November 28, 1984

# ABSTRACT

Mixtures of carbon dioxide and some hydrocarbons will form liquid-liquid-vapor equilibrium at temperatures and pressures considered reservoir conditions. So that carbon dioxide miscible flooding may be properly simulated, accurate modeling of the phase behavior is necessary. An algorithm has been developed which is capable of predicting the two-phase and liquid-liquid-vapor equilibrium shown by $CO_2$-hydrocarbon mixtures. The algorithm conducts phase stability analysis calculations in series with flash calculations using the Peng-Robinson equation of state to model fluid behavior of all phases present.

Example calculations were made of $CO_2$ mixtures with normal alkanes which show liquid-vapor, liquid-liquid, and also liquid-liquid-vapor behavior. An example calculation was also made for a mixture of $CO_2$ with a reservoir oil.

## TABLE OF CONTENTS

ACKNOWLEDGMENTS . . . . . . . . . . . . . . . . . . . . iii

ABSTRACT . . . . . . . . . . . . . . . . . . . . . . . iv

TABLE OF CONTENTS . . . . . . . . . . . . . . . . . . . v

LIST OF TABLES . . . . . . . . . . . . . . . . . . . . vii

LIST OF FIGURES . . . . . . . . . . . . . . . . . . . . ix

CHAPTER

1. Introduction . . . . . . . . . . . . . . . . . . 1

2. Multi-Phase Equilibria of $CO_2$-Hydrocarbon Mixtures . . . . . . . . . . . . . . . . . . . 5

3. Equations For Multi-Phase Equilibria . . . . . 19
    - 3.1 Thermodynamic Description of Phase Equilibria . . . . . . . . . . . . . . 19
    - 3.2 Material Balance Equations . . . . . 27
    - 3.3 Conditions for a Stable Phase Equilibrium Solution . . . . . . . . 30

4. Phase Stability Analysis . . . . . . . . . . . 37

5. Solution Algorithm . . . . . . . . . . . . . . 43
    - 5.1 Overall Logic of Algorithm . . . . . 43
    - 5.2 Initial Two-Phase Stability Analysis . . . . . . . . . . . . . . . 45
    - 5.3 Two-Phase Flash Calculation . . . . . 49
        - 5.3.1 Generation of Starting Values . . . . . . . . . . . 49
        - 5.3.2 Solution Methods of Flash Equations . . . . . . . . . 52
    - 5.4 Stability Analysis Test for an Additional Phase . . . . . . . . . . 57
    - 5.5 Three-Phase Flash Calculation . . . . 59
        - 5.5.1 Starting Values . . . . . . 59
        - 5.5.2 Three-Phase Flash . . . . . 60

6. Results . . . . . . . . . . . . . . . . . . . 64
    6.1 $CO_2$-n-Butane Binary Mixture . . . . . 64
    6.2 $CO_2$-n-Decane Binary Mixture . . . . . 71
    6.3 $CO_2$-$CH_4$-n-$C_{16}$ Ternary Mixture . . . . 87
    6.4 $CO_2$-$C_3H_8$-n-$C_{16}$ Ternary Mixture . . . 103
    6.5 $CO_2$-Crude Oil Mixture . . . . . . . . . 112
    6.6 Comparison of Numerical Methods . . . 121
    6.7 Problems Encountered with this Algorithm . . . . . . . . . . . . . . . 126

7. Summary and Conclusions . . . . . . . . . . . 129
    7.1 Summary . . . . . . . . . . . . . . . . . 129
    7.2 Conclusions . . . . . . . . . . . . . . . 130
    7.3 Recommendations . . . . . . . . . . . . 131

NOMENCLATURE . . . . . . . . . . . . . . . . . . . . . 133

APPENDIX A Derivation of Phase Stability Criteria . . 135

APPENDIX B Listing of Source Code . . . . . . . . . . 140

BIBLIOGRAPHY . . . . . . . . . . . . . . . . . . . . . 194

VITA

# LIST OF TABLES

| Table | | Page |
|---|---|---|
| 6.1 | Component critical constants and acentric factors used in example calculations . . . . . . | 67 |
| 6.2 | Binary interaction parameters used in example calculations . . . . . . . . . . . . . . . . . . . | 68 |
| 6.3 | Calculated equilibrium phase compositions for $CO_2$ and n-decane at 220°F and 2300 psia for two values of the binary interaction coefficient . | 77 |
| 6.4 | Calculated phase densities and compressibility factors for $CO_2$ and n-decane at 220°F and 2300 psia for two values of the binary interaction coefficient . . . . . . . . . . . . . . . . | 79 |
| 6.5 | Phase compositions for $CO_2$, $CH_4$, and n-$C_{16}$ mixture at 70°F and 945 psia . . . . . . . . . . | 90 |
| 6.6 | Phase properties and phase distribution for $CO_2$, $CH_4$, and n-$C_{16}$ mixture at 70°F and 945 psia . . . . . . . . . . . . . . . . . . . . . | 90 |
| 6.7 | Phase compositions for $CO_2$, $CH_4$, and n-$C_{16}$ mixture at 70°F and 1000 psia . . . . . . . . . | 91 |
| 6.8 | Phase properties and phase distribution for $CO_2$, $CH_4$, and n-$C_{16}$ at 70°F and 1000 psia . . . | 91 |
| 6.9 | Phase compositions and densities for the $CO_2$-$CH_4$ binary mixture at 70°F and 900 psia . . . . | 94 |
| 6.10 | Phase compositions and densities for the $CO_2$, $CH_4$, and n-$C_{16}$ ternary at 70°F and 2000 psia . . | 99 |
| 6.11 | Phase compositions and densities for the $CO_2$, $CH_4$, and n-$C_{16}$ ternary at 70°F and 2000 psia . . | 99 |
| 6.12 | Phase compositions and densities for the mixture of $CO_2$, $C_3H_8$, and n-$C_{16}$ at 70°F and 700 psia . . . . . . . . . . . . . . . . . . . . . | 108 |

Table Page

6.13 Fluid composition analysis of recombined reservoir oil C2 taken from Turek et al. [T2] . 113

6.14 Constants used in the general property correlation, Equation 6.5.1 . . . . . . . . . . . . . 115

6.15 Constants used in the binary interaction coefficient correlation, Equation 6.5.3 . . . . 117

6.16 Calculated phase compositions and properties for a $CO_2$ and reservoir oil mixture at 542.5°R and 1060 psia . . . . . . . . . . . . . . . . . 120

6.17 Iterations and execution times for solution of phase stability equations for a ten component mixture . . . . . . . . . . . . . . . . . . . . 122

6.18 Iterations and execution times for solution of phase stability equations for a ternary mixture . . . . . . . . . . . . . . . . . . . . 124

6.19 Execution times required for a phase equilibria solution, including stability analysis . . . . . 125

# LIST OF FIGURES

Fig. Page

2.1 Pressure-temperature diagram for the mixture of carbon dioxide and normal decane. Experimental data taken from Kulkarni, Zarah, Luks, and Kohn [K1] . . . . . . . . . . . . . . . . . 6

2.2 Location of the three-phase region in the mixture of $CO_2$, $C_3H_8$ and n-$C_{16}$. Experimental data is from Meldrum and Nielsen [M4] . . . . . 8

2.3 Schematic of ternary diagram and the location of the phase envelopes for the mixture of $CO_2$, $CH_4$ and n-$C_{16}$ at a fixed temperature and pressure . . . . . . . . . . . . . . . . . . . . 10

2.4 Location of the three-phase region with changes in pressure. Data taken from Meldrum and Nielsen [M4] at a fixed temperature of 70°F . . 11

2.5 Schematic pressure-overall composition diagram for the mixture of $CO_2$, $C_3H_8$ and n-$C_{16}$ at a fixed temperature . . . . . . . . . . . . . . . 13

2.6 Schematic pressure-overall composition diagram for the mixture of $CO_2$, $CH_4$ and n-$C_{16}$ at a fixed temperature . . . . . . . . . . . . . . . 14

2.7 Pressure-composition diagram for reservoir oil A with $CO_2$ at 130°F, taken from [S2] . . . . 16

2.8 Pressure-composition diagram for recombined reservoir oil B with $CO_2$ at 94°F, taken from [S1] . . . . . . . . . . . . . . . . . . . . . . 17

3.1 Molar Gibbs free energy of mixing curve for a binary mixture at pressure P, and temperature $T_1$ . . . . . . . . . . . . . . . . . . . . . . . 32

3.2 Molar Gibbs free energy of mixing curve for a binary mixture at pressure P, and temperature $T_2$ . . . . . . . . . . . . . . . . . . . . . . . 33

Fig. Page

3.3 Molar Gibbs free energy of mixing curve for a binary mixture at pressure P, and temperature $T_3$ . . . . . . . . . . . . . . . . . . . . . . 35

5.1 Overall procedure for multi-phase flash algorithm . . . . . . . . . . . . . . . . . . . . 44

5.2 Schematic for successive substitution method of solution of phase stability analysis . . . . . . 46

5.3 Flowchart for accelerated successive substitution method . . . . . . . . . . . . . . . . . . 54

6.1 Computed liquid-vapor phase envelope for the $CO_2$ and n-butane binary mixture at several temperatures . . . . . . . . . . . . . . . . . . . . 65

6.2 Pressure versus composition diagram for the $CO_2$ and n-butane mixture. Comparison of calculated and measured phase composition at two temperatures . . . . . . . . . . . . . . . . . . . . . 70

6.3 Pressure versus composition diagram for the $CO_2$ and n-butane mixture. Comparison of calculated and measured phase composition at two temperatures . . . . . . . . . . . . . . . . . . . . . 72

6.4 Molar Gibbs free energy of mixing curve for a binary mixture of $CO_2$ and n-decane at 100°F and 1300 psia . . . . . . . . . . . . . . . . . . . . 73

6.5 Computed phase equilibria for the binary mixture of $CO_2$ and n-decane showing liquid-liquid and liquid-vapor behavior . . . . . . . . . . . 74

6.6 Pressure-composition diagram for the $CO_2$ and n-decane binary mixture. Calculated phase envelope at higher temperatures for a binary interaction coefficient of 0.115 . . . . . . . . 76

6.7 Pressure versus composition diagram for the $CO_2$ and n-decane mixture. Comparison of calculated and measured phase composition at two temperatures . . . . . . . . . . . . . . . . . . . . . 80

Fig. Page

6.8 Pressure-composition diagram for the mixture of $CO_2$ and n-decane at 100°F. Calculated phase envelope for several values of the binary interaction coefficient . . . . . . . . . . . . 82

6.9 Pressure-composition diagram for the mixture of $CO_2$ and n-decane at 100°F. Calculated phase envelope for several values of the binary interaction coefficient . . . . . . . . . . . . 83

6.10 Pressure-composition diagram for the $CO_2$ and n-decane mixture at 100°F. Calculated phase envelope for two values of the n-decane acentric factor . . . . . . . . . . . . . . . . . 84

6.11 Pressure-composition diagram for the $CO_2$ and n-decane mixture at 100°F. Calculated phase envelope for two values of the n-decane critical temperature . . . . . . . . . . . . . . . . . 85

6.12 Pressure-composition diagram for the $CO_2$ and n-decane mixture at 100°F. Calculated phase envelope for two values of the n-decane critical pressure . . . . . . . . . . . . . . . . . . . 86

6.13 Computed phase distribution versus pressure for the mixture of $CO_2$, $CH_4$, and n-$C_{16}$ at 70°F and for fixed overall composition . . . . . . . . . 88

6.14 Computed ternary diagram for the mixture of $CO_2$, $CH_4$, and n-$C_{16}$ at 70°F and 700 psia . . . . 92

6.15 Computed ternary diagram for the mixture of $CO_2$, $CH_4$, and n-$C_{16}$ at 70°F and 900 psia . . . . 93

6.16 Computed ternary diagram for the mixture of $CO_2$, $CH_4$, and n-$C_{16}$ at 70°F and 1000 psia . . . 96

6.17 Computed ternary diagram for the mixture of $CO_2$, $CH_4$, and n-$C_{16}$ at 70°F and 1000 psia . . . 97

6.18 Computed ternary diagram for the mixture of $CO_2$, $CH_4$, and n-$C_{16}$ at 70°F and 2000 psia . . . 98

Fig. Page

6.19 Movement of the computed three-phase region as the pressure varies for the mixture of $CO_2$, $CH_4$, and n-$C_{16}$ at 70°F . . . . . . . . . . . . . . 101

6.20 Pressure-composition diagram for a mixture of $CO_2$ with $CH_4$ and n-$C_{16}$ at 70°F. The ratio of $CH_4$ to n-$C_{16}$ is fixed at 30:70 . . . . . . . . . . 102

6.21 Comparison of the experimental and computed location of the three-phase region for the mixture of $CO_2$, $C_3H_8$ and n-$C_{16}$ at 70°F . . . . . 104

6.22 Computed ternary diagram for the mixture of $CO_2$, $C_3H_8$, and n-$C_{16}$ at 70°F and 500 psia . . . 105

6.23 Computed ternary diagram for the mixture of $CO_2$, $C_3H_8$, and n-$C_{16}$ at 70°F and 700 psia . . . 107

6.24 Computed ternary diagram for the mixture of $CO_2$, $C_3H_8$, and n-$C_{16}$ at 70°F and 795 psia . . . 109

6.25 Computed ternary diagram for the mixture of $CO_2$, $C_3H_8$, and n-$C_{16}$ at 70°F and 1000 psia . . . 110

6.26 Movement of the computed three-phase region as the pressure varies for the mixture of $CO_2$, $C_3H_8$, and n-$C_{16}$ at 70°F . . . . . . . . . . . . 111

6.27 Computed phase distribution versus pressure for the $CO_2$-reservoir oil mixture. The overall $CO_2$ mole fraction is 75% . . . . . . . . . . . . . . . 119

# CHAPTER 1

## Introduction

It has been found [K1][S1] that mixtures of carbon dioxide with some hydrocarbons will form up to four coexisting phases at equilibrium. This can occur at temperatures and pressures considered as reservoir conditions. So that carbon dioxide miscible flooding may be properly simulated, accurate modeling of the complex phase behavior is necessary. In this study, an algorithm is developed which is capable of modeling phase behavior shown by carbon dioxide-hydrocarbon mixtures while at the same time being suitable for use in compositional simulators.

The behavior which must be modeled is the equilibrium of phases which may have mass transfer between them. For the purpose of this study, it is assumed that the species comprising the phases are non-reacting. Hence there are no chemical reactions. Also, the phases considered are non-aqueous. That is, they contain no water.

Much of the previous work was centered on the equilibria between a liquid phase and a vapor phase. One of the common methods of phase equilibria calculations utilizes an equation of state to model fluid behavior. The techniques are iterative and a variety of methods are used to find the equilibrium solution.

One method given by Fussell and Yanosik [F1] was called the Minimum-Variable-Newton-Raphson technique. The thermodynamic criterion for phase equilibria provided one equation for each component present in the mixture. Then, given a fixed temperature, pressure and overall composition, the equilibrium phase compositions and relative amounts of each phase could be found by solving for nc independent variables, where nc is the number of components present in the mixture. The term "minimum variable" came about because under the above conditions, only nc variables were needed to specify the state of the system completely. The Newton-Raphson method for solution of simultaneous, non-linear equations was then applied to find the phase equilibrium solution. In their work, Fussell and Yanosik used the Redlich-Kwong [R2] equation of state.

One drawback of the Newton-Raphson method is that a Jacobian matrix must be calculated at each iteration. A simple alternative is the successive substitution method, but this method shows poor convergence near the critical point. A method was devised by Nghiem and Aziz [N2] which started out using successive substitution. At the point in the iteration sequence where a set of criteria were met, a switch was made from the successive substitution method to

Powell's [P3] method, which is a quasi-Newton technique. This switching method utilized the simplicity of the successive substitution method with the strong convergence characteristics of Powell's method.

Mehra, Heidemann and Aziz [M1][M3] also used a switching technique, but applied an acceleration of the successive substitution method to flash calculations. This ACSS [M2] method was a relatively simple modification to the successive substitution method yet provided faster convergence overall and vastly improved convergence characteristics near the critical region. In addition, this method was extended to three equilibrium phases.

Published three-phase flash calculation techniques include the above by Mehra et al., an extension of Minimum-Variable-Newton-Raphson methods by Lynne Fussell [F2], and an algorithm developed by Risnes and Dalen [R2]. Like Mehra's technique, the method proposed by Risnes is performed in a stepwise fashion. Initially a two-phase flash is done. At this point, a check is made to label the types of phases present--either liquid or vapor. Next, a droplet is formed from one of the phases to verify if the droplet will grow to a converged composition or disappear. If the droplet does grow, a phase is added and a three-phase flash calculation is conducted.

One of the difficulties of multi-phase flash calculations is determining the number of phases which should exist at equilibrium. Michelsen [M6][M7] developed a phase stability analysis technique which can be used in a stepwise fashion for multi-phase calculations. This technique for multi-phase, multi-component phase equilibria stability calculations was used in this study. Nghiem and Li [N3] incorporated Michelsen's scheme in their most recent paper presented at the Fall SPE meeting just after the completion of this study.

In Chapter 2, several examples of the complexity of the phase behavior of carbon dioxide—hydrocarbon mixtures are described. Chapter 3 presents the thermodynamic criteria for phase equilibrium along with other equations used in the flash calculation. Chapter 4 describes the phase stability analysis method as used in the algorithm. A detailed description of the entire algorithm is given in Chapter 5 and several results are presented in Chapter 6. Finally, Chapter 7 contains conclusions and recommendations for further study.

## CHAPTER 2

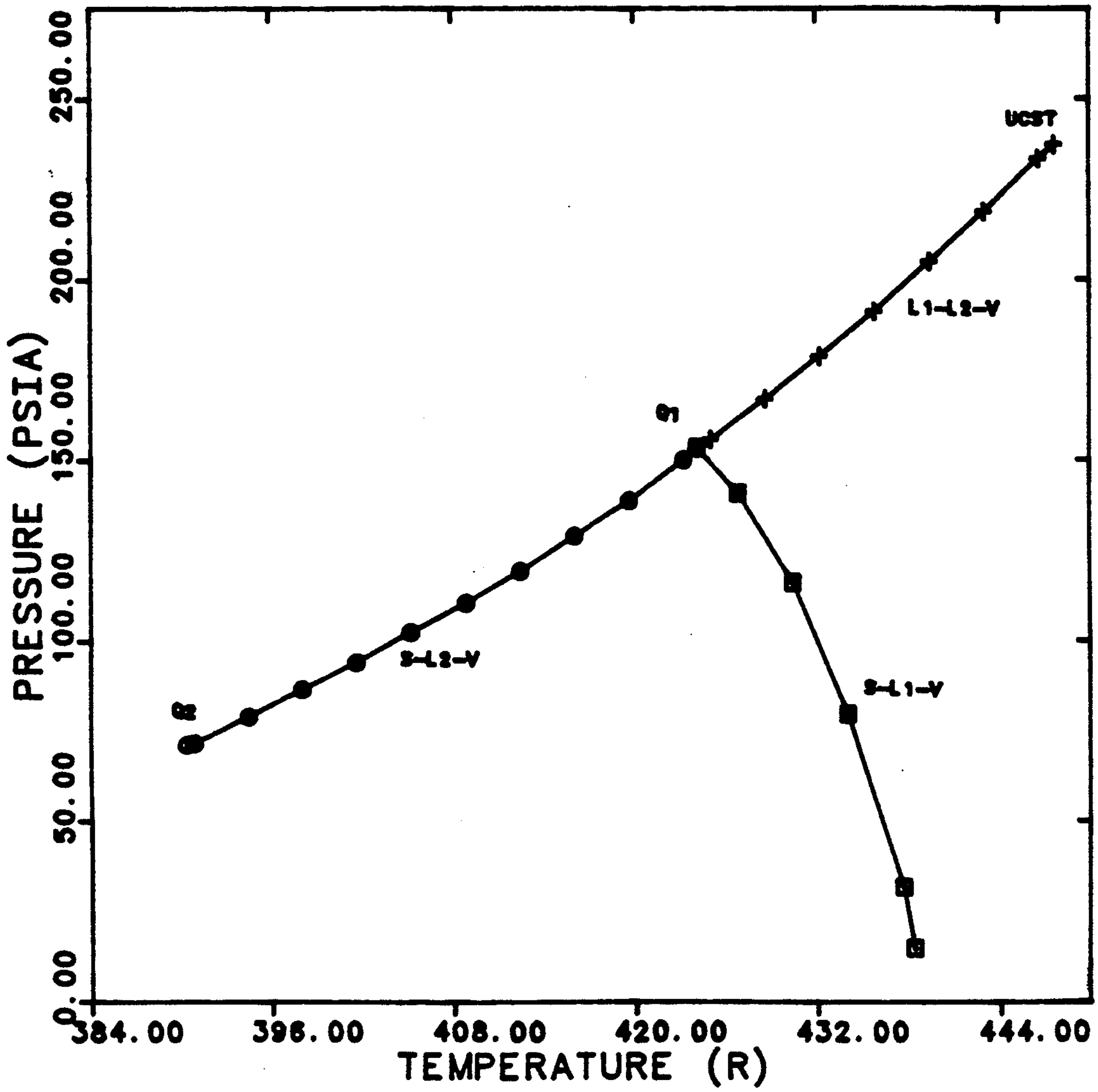
## Multi-Phase Equilibria of $CO_2$-Hydrocarbon Mixtures

Phase behavior is an important factor in the efficiency of oil displacement by carbon dioxide ($CO_2$) [M5][G1]. The behavior of $CO_2$ with hydrocarbon mixtures can become quite complex. Several examples are presented here.

At higher temperatures, above about 120°F, primarily liquid-vapor behavior has been observed for $CO_2$-hydrocarbon mixtures [O3]. However, at lower temperatures, liquid-liquid equilibria has been found as well as liquid-liquid-vapor behavior. In addition, precipitation of a solid has been reported [S1][G1][S2] for a wide range of temperatures and pressures.

An examination of the phase behavior with hydrocarbons begins with binary mixtures of $CO_2$ and normal alkanes. For the alkanes methane through hexane, only vapor-liquid equilibria is found. When the alkane is heptane or higher carbon number, liquid-liquid and liquid-liquid-vapor behavior have been observed in addition to vapor-liquid equilibria depending on the temperature and pressure [O3]. Figure 2.1 illustrates a pressure-temperature diagram for the carbon dioxide-normal decane binary mixture, as investigated

FIGURE 2.1 PRESSURE–TEMPERATURE DIAGRAM FOR THE MIXTURE OF CARBON DIOXIDE AND NORMAL DECANE. EXPERIMENTAL DATA TAKEN FROM KULKARNI, ZARAH, LUKS, AND KOHN[K1].

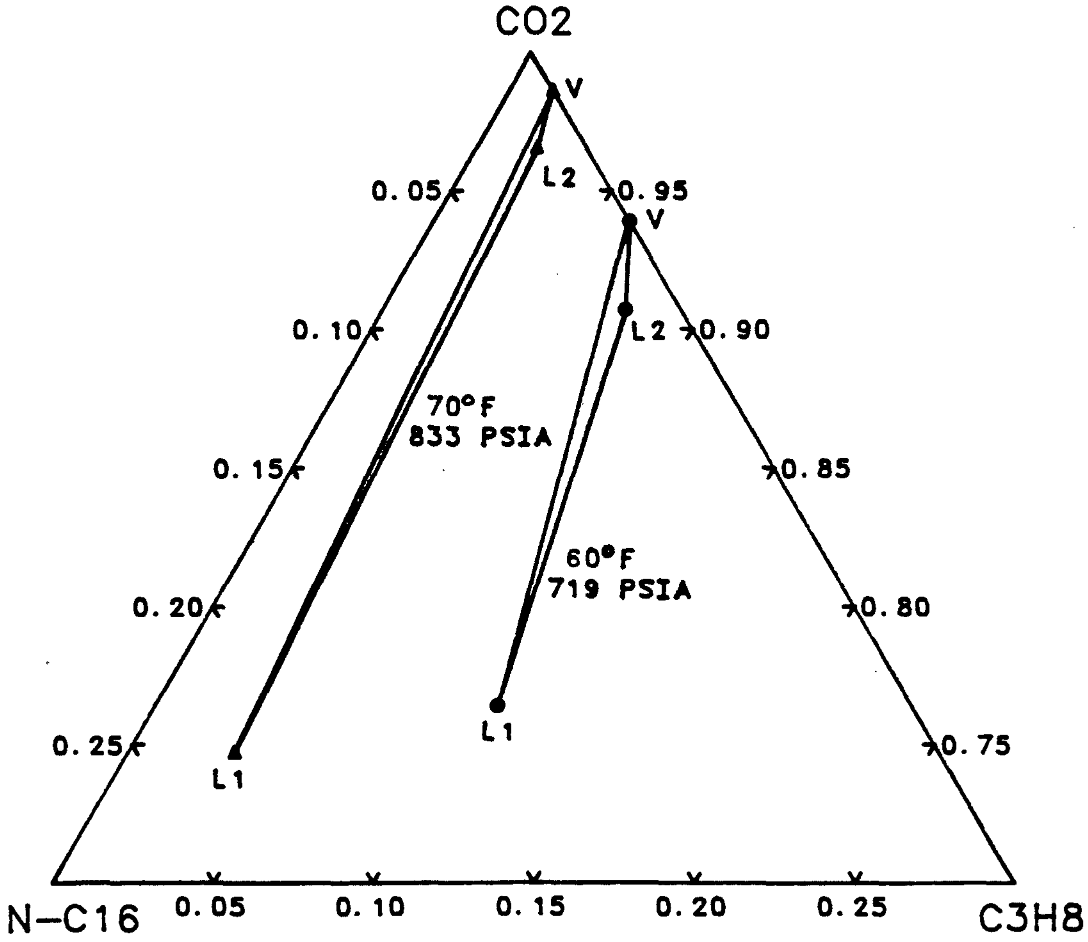


by Kulkarni, Zarah, Luks, and Kohn [K1]. The highest temperature reported for liquid-liquid-vapor equilibrium was about -12°F, where the three-phase curve ends in a critical point. As the alkane increases in carbon number, the maximum temperature for liquid-liquid-vapor equilibrium also increases. This three-phase equilibria may occur at temperatures greater than the critical temperature of carbon dioxide when the alkane is $C_{13}$ or heavier. Experimental results of $CO_2$-alkane binary phase behavior studies are available in the literature for many normal alkanes, including some equilibrium phase composition data [P2][O1][I1][S4].

Ternary mixtures of carbon dioxide and normal alkanes have also been studied. At low pressures, vapor-liquid equilibrium is found over a wide range of temperatures. At low temperatures, when one of the hydrocarbons is heavy enough, liquid-liquid and liquid-liquid-vapor equilibrium are observed at higher pressures. Liquid-liquid behavior often continues to very high pressures.

Meldrum and Nielsen [M4] presented phase composition results for the ternary mixture of carbon dioxide, propane and normal-hexadecane ($n-C_{16}$). The mixture was examined at 60°F and 70°F for several pressures. Liquid-liquid-vapor equilibrium was observed and some of their data are shown in Figure 2.2.

FIGURE 2.2 LOCATION OF THE THREE-PHASE REGION IN THE MIXTURE OF CO2, C3H8 AND N-C16. EXPERIMENTAL DATA IS FROM MELDRUM AND NIELSEN [M4].



For a ternary diagram such as this, pressure and temperature are fixed. The three-phase region is a triangle whose corner points are invariant. That is, when the overall composition changes but still remains within the three-phase region, only the relative amounts of each phase will change. The equilibrium phase compositions do not change. The situation is analogous to moving along a tie-line in two-phase equilibria.

In Figure 2.2, the three-phase region is shown. However, a two-phase region exists adjacent to each side of the three-phase triangle which is not shown, since Meldrum and Nielsen did not present any additional two-phase data for the three-phase cases. Of the three two-phase regions, two are liquid-vapor and the third is liquid-liquid. A schematic of this is shown for carbon dioxide,methane and normal hexadecane in Figure 2.3.

In Figure 2.4, the location of the three-phase region is shown as pressure varies. As the pressure increases, the three-phase region moves toward the $CO_2$-n-$C_{16}$ side of the ternary diagram. Eventually, the three-phase region will disappear when the pressure is high enough. This can be contrasted with the ternary system of carbon dioxide, methane and normal hexadecane. When pressure is increased, the three-phase region moves toward higher methane concentrations.

FIGURE 2.3 SCHEMATIC OF TERNARY DIAGRAM AND THE LOCATION OF THE PHASE ENVELOPES FOR THE MIXTURE OF CO2, CH4 AND N-C16 AT A FIXED TEMPERATURE AND PRESSURE.

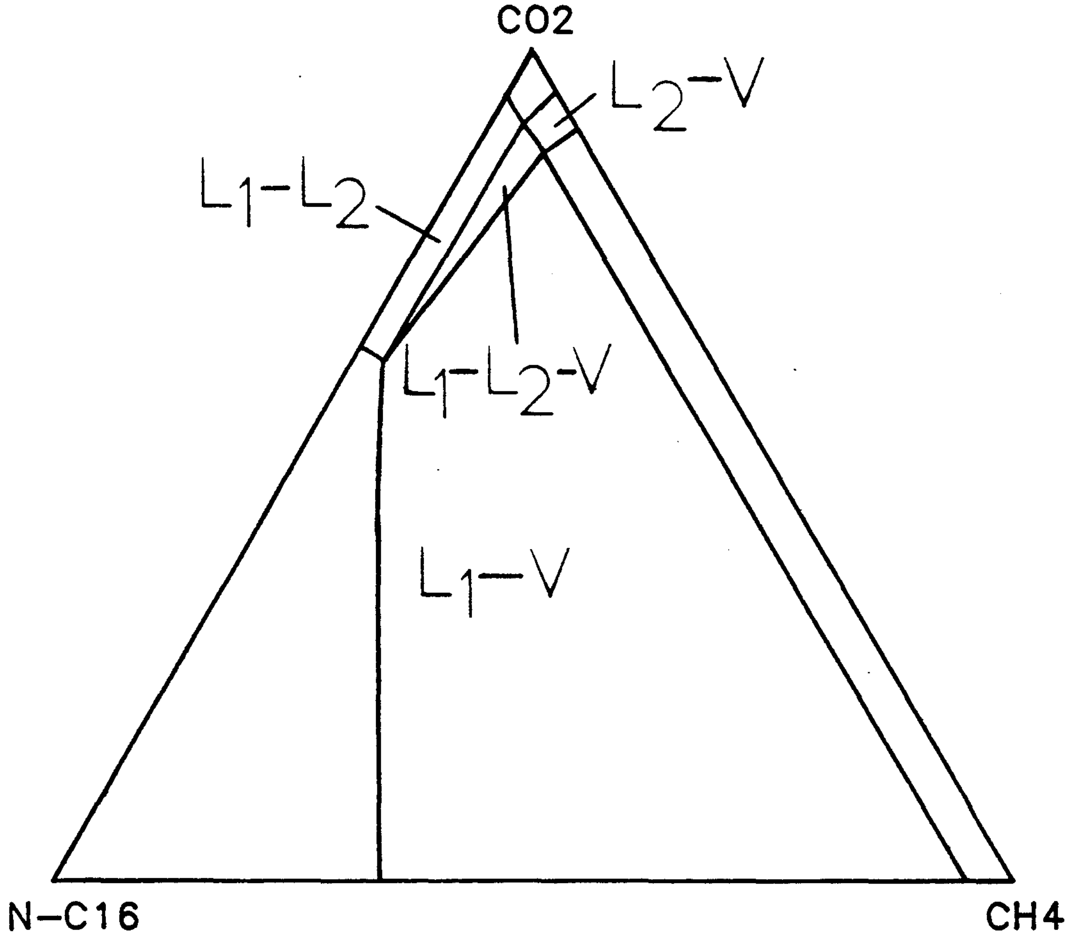
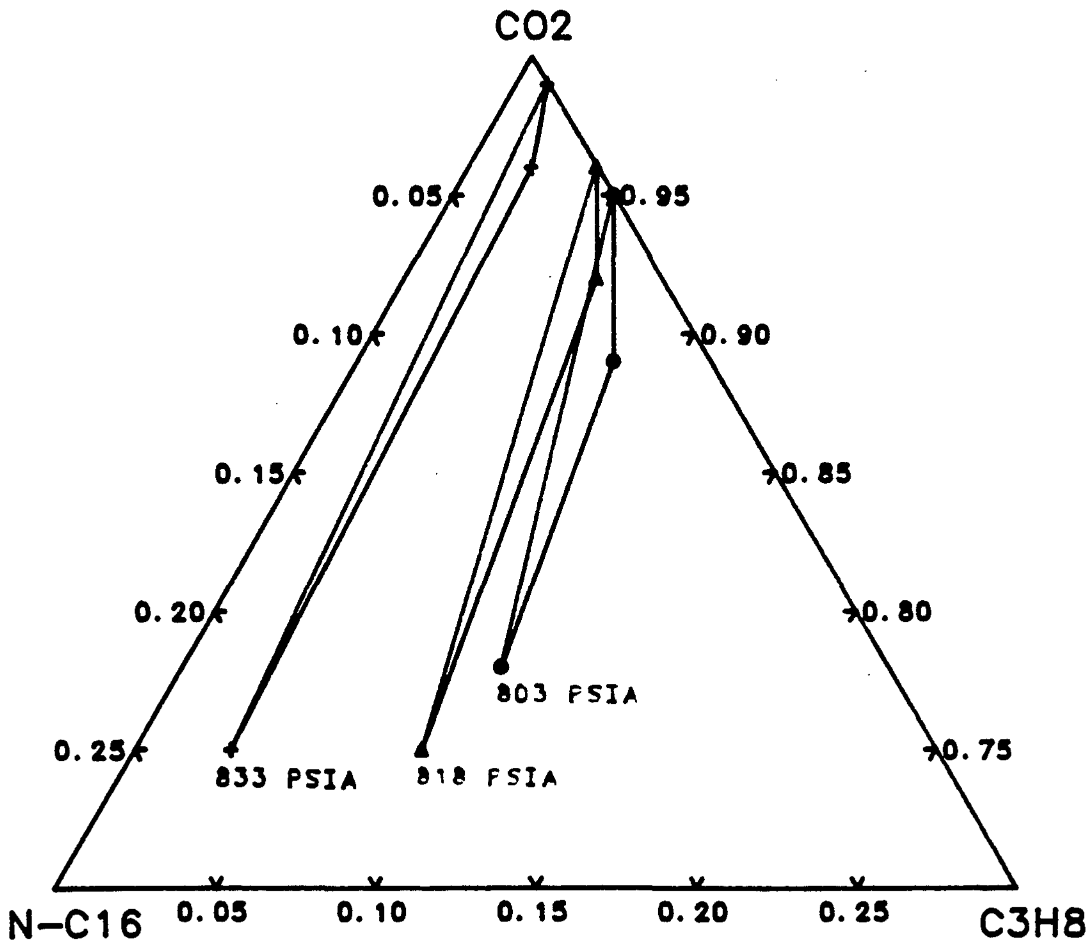


FIGURE 2.4 LOCATION OF THE THREE-PHASE REGION WITH CHANGES IN PRESSURE. DATA TAKEN FROM MELDRUM AND NIELSEN [M4] AT A FIXED TEMPERATURE OF 70°F.



The comparison between these two mixtures may be seen in another context by making a pressure-overall composition diagram for each. That is, assume a fixed ratio of n-hexadecane to methane or propane; for example, 70:30. At a fixed temperature and pressure, carbon dioxide may be added to the mixture and the number of phases formed be observed. When these dilution lines are examined for several pressures with the temperature fixed and constant, the results may be plotted on a pressure-overall composition diagram [02]. Note that the regions outlined in the diagram do not give equilibrium phase compositions. For an overall composition which lies in a particular region for a given pressure, the diagram only indicates the number of phases that will form at equilibrium. Figures 2.5 and 2.6 are schematics for the $CO_2$-propane-n-$C_{16}$ and $CO_2$-methane-n-$C_{16}$ mixtures, respectively. Here the pressure versus overall mole fraction of carbon dioxide is plotted.

The movement of the three-phase region in the ternary diagram as pressure changes may be seen by the slope of the three-phase envelope. As pressure increases, the three-phase region of the $CO_2$-methane-n-$C_{16}$ mixture moves toward higher concentrations and lower carbon dioxide concentrations. Therefore, the three-phase envelope slopes downward. The $CO_2$-propane-n-$C_{16}$ mixture behaves in a different manner. As pressure increases, the three-phase

FIGURE 2.5 SCHEMATIC PRESSURE-OVERALL COMPOSITION DIAGRAM FOR THE MIXTURE OF CO2, C3H8 AND N-C16 AT A FIXED TEMPERATURE.

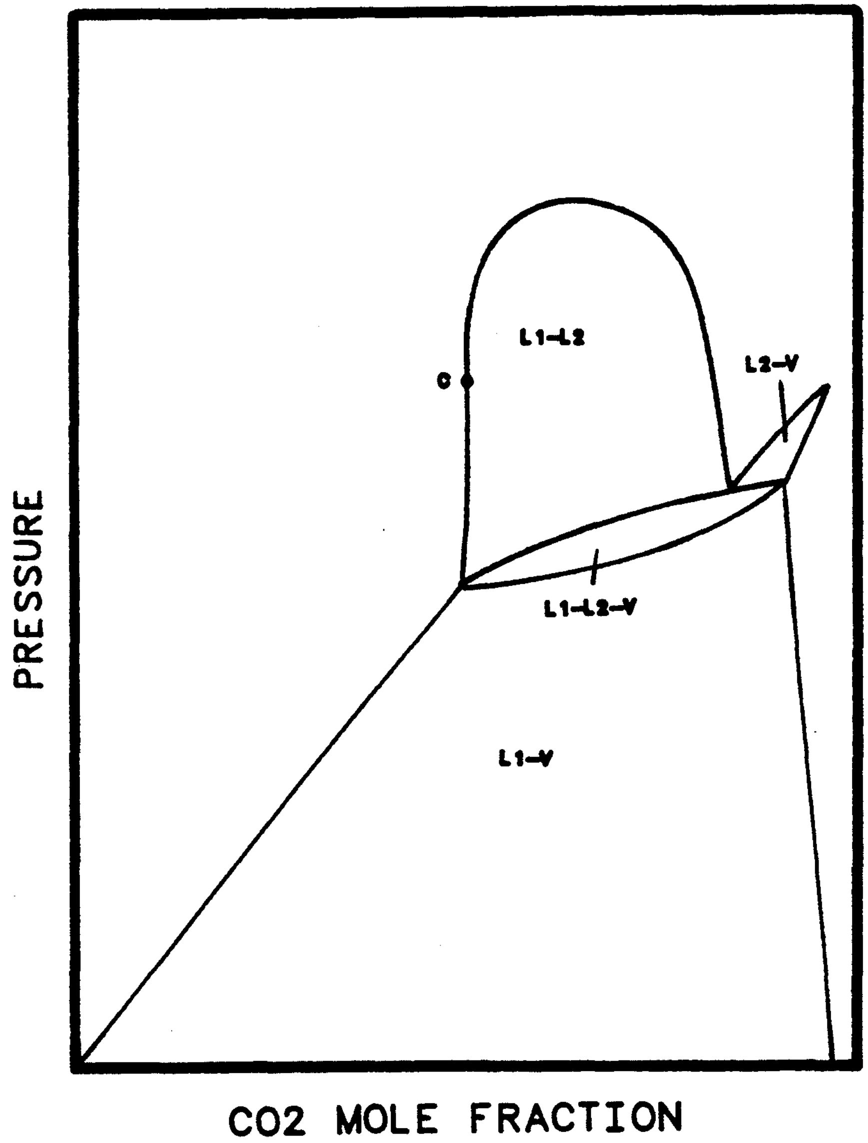
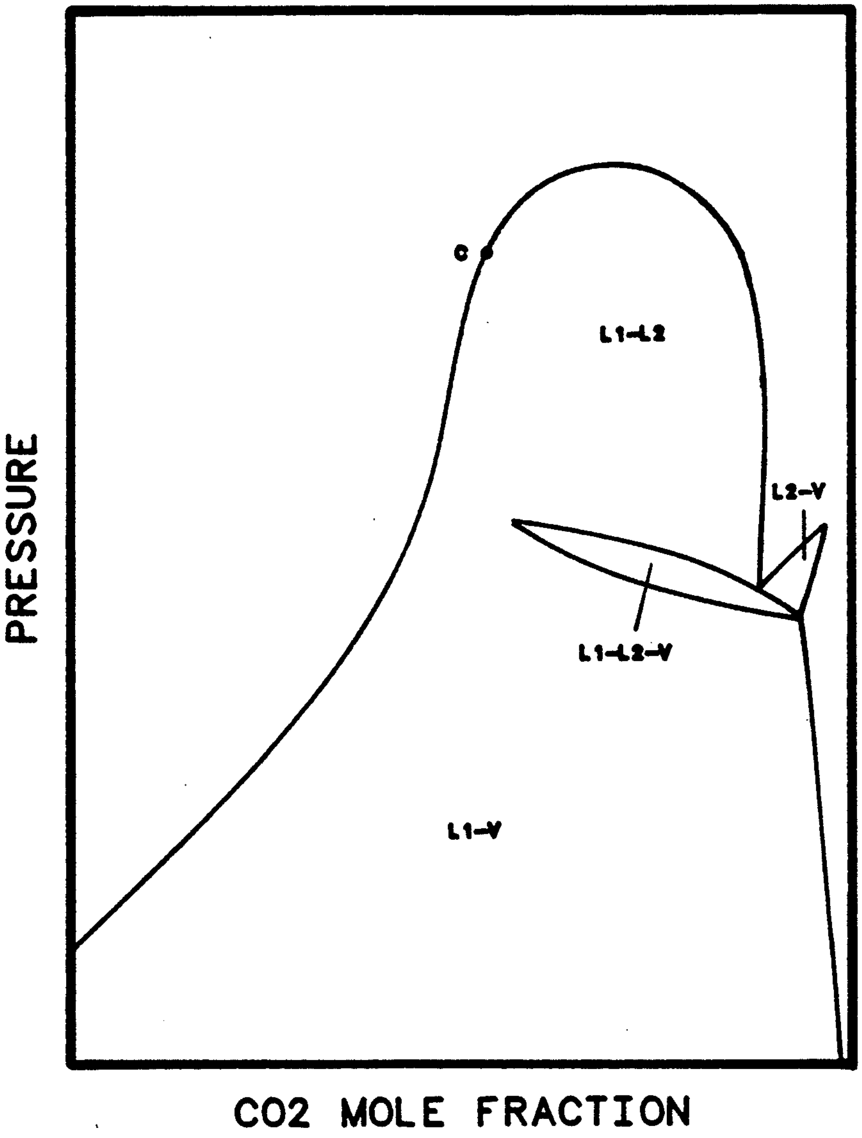


FIGURE 2.6 SCHEMATIC PRESSURE–OVERALL COMPOSITION DIAGRAM FOR THE MIXTURE OF CO2, CH4 AND N–C16 AT A FIXED TEMPERATURE.



region moves toward higher $CO_2$ concentrations, and as a result, the three-phase envelope slopes upward. Orr [O4] has classified the behavior of the $CO_2$-$C_3$-n-$C_{16}$ mixture as Type II and the behavior of the $CO_2$-$C_1$-n-$C_{16}$ mixture as Type I. $CO_2$ and crude oil mixtures also show this behavior.

Simon [S2] reported the phase behavior properties of two reservoir oils with various amounts of carbon dioxide. The experiments were conducted at a temperature of 130°F for oil A and 255°F for oil B. Vapor-liquid equilibria was reported, which could be expected at these temperatures considering the behavior of the $CO_2$-alkane mixtures. A solid precipitate was also reported for mixtures containing over 60 mole per cent of $CO_2$. A pressure-composition diagram for oil A is presented in Figure 2.7.

Shelton and Yarborough [S1] found liquid-liquid and liquid-vapor phase behavior in their experiments with carbon dioxide and a reservoir oil. The test temperature was at 94°F. They also reported the formation of a tar-like resin precipitate for several mixtures of varying $CO_2$ content. The phase behavior results showed that a liquid-liquid equilibrium existed for all eight mixtures with $CO_2$. The lowest $CO_2$ content was a mole fraction of .58. For some mixtures, the liquid-liquid equilibrium persisted up to more than 9000 psia. A pressure-composition diagram is shown for $CO_2$ with oil B in Figure 2.8

FIGURE 2.7 PRESSURE-COMPOSITION DIAGRAM FOR RESERVOIR OIL A WITH CO2 AT 130°F, TAKEN FROM [S2].

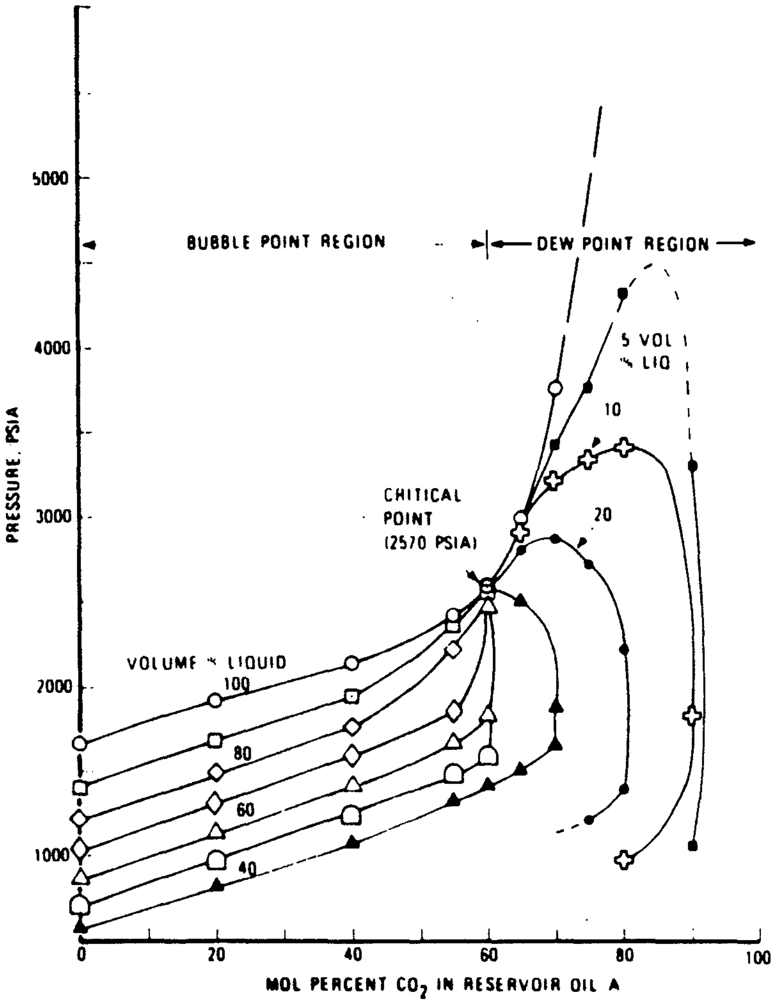
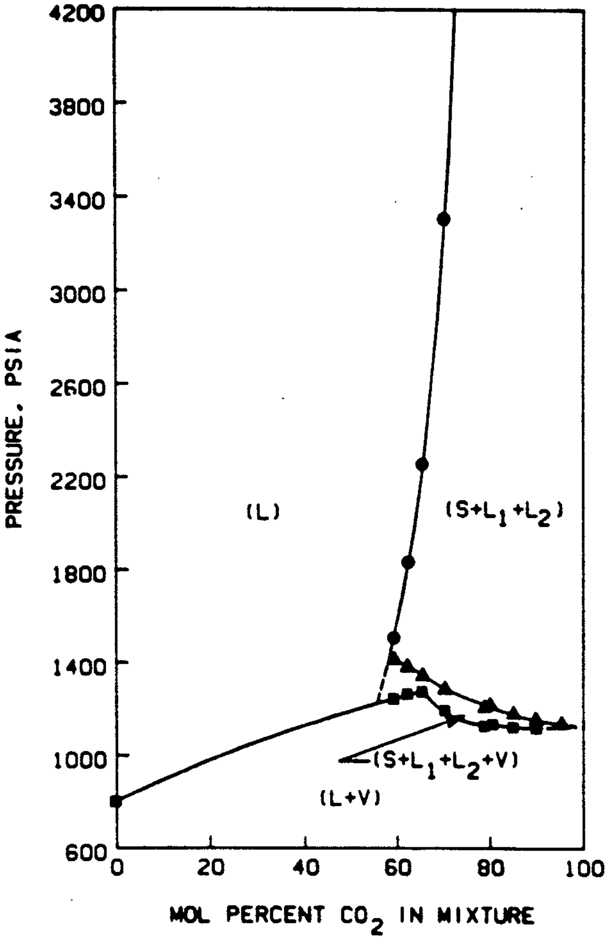


FIGURE 2.8 PRESSURE–COMPOSITION DIAGRAM FOR RECOMBINED RESERVOIR OIL B WITH CO2 AT 94°F, TAKEN FROM [S1].



Finally, Turek, Metcalfe and Fishback [T1] reported phase composition data for carbon dioxide mixtures with several reservoir oils. In their study, they found that mixtures containing less than about 60 mole per cent carbon dioxide show conventional vapor-liquid equilibria. When the temperature was below about 110°F, a three-phase liquid-liquid-vapor region was found for mixtures containing high concentrations of carbon dioxide. Liquid-liquid-vapor behavior was not found above 110°F. Liquid-liquid equilibria was observed for mixtures containing more than 70 mole per cent $CO_2$ at temperatures below 110°F and at elevated pressures. No liquid-liquid critical point was found up to 3000 psia. Results from other researchers show a liquid-liquid critical at much lower pressures, indicating, perhaps, a strong relationship to oil composition.

As shown by the behavior of $CO_2$ and alkanes, the phase behavior is strongly affected by the light components present in the reservoir oil. It was found that oils containing less of the light components had a greatly decreased pressure range over which three phases were formed.

# CHAPTER 3

## Equations For Multi-Phase Equilibria

In this chapter the equations useful for calculation of multi-phase, multi-component equilibrium will be presented. In the first section, the equations for the thermodynamic criteria for phase equilibria are given. Also in this section, the Peng-Robinson [P1] equation of state is described. In the second section, the equations for the material balances as used in the flash calculation are stated. In the final section the conditions for a stable phase equilibrium solution are discussed.

### 3.1 Thermodynamic Description of Phase Equilibria

For a system capable of undergoing a process in which the composition may change and if the temperature and pressure are uniform throughout, the condition for equilibrium is given by

$$\mu_i{}^a = \mu_i{}^b = \ldots = \mu_i{}^m$$

$$\text{for } i = 1,\ 2,\ \ldots,\ nc \qquad (3.1.1)$$

where $\mu_i^a$ is the chemical potential of component i in phase a, $\mu_i^b$ is the chemical potential of component i in phase b, etc. This equation states that the system is at equilibrium if the chemical potential of a component is equal in every phase present at equilibrium. There are several ways that the chemical potential may be defined. Thus for a process that occurs at constant temperature and pressure, the component chemical potential is given by

$$\mu_i = \left( \frac{\partial (nG)}{\partial n_i} \right)_{T,\, P,\, n_j} \qquad (3.1.2)$$

where nG is the total Gibbs free energy and $n_i$ is the mole number of component i.

A thermodynamic property which is defined in terms of the chemical potential is the fugacity, f. The fugacity is useful for calculations of phase equilibrium. Fugacity and chemical potential are related by

$$\mu_i = \mu_i^0 + RT \ln \frac{f_i}{f_i^0} \qquad (3.1.3)$$

$f_i$ is the fugacity of component i in a mixture and $\mu_i$ is the chemical potential of component i in the mixture. Also, $\mu_i^0$ is the standard state chemical potential and is a function of temperature and pressure only.

It can be shown that a condition for equilibrium equivalent to Equation 3.1.1 in terms of component fugacities is given by

$$f_i^a = f_i^b = \dots = f_i^m$$

$$\text{For } i = 1, 2, \dots, nc \qquad (3.1.4)$$

which states that the fugacity of a component must be equal in all phases present at equilibrium. Equation 3.1.4 is the form of the thermodynamic criterion which will be used for phase equilibria calculations.

One method used for phase equilibria calculations is to compute the component fugacities from an equation of state. This is done for each phase present regardless of whether the phase is a vapor or liquid. For the most part, equations of state are unable to describe the volumetric properties of condensed phases accurately, that is, liquid, solid, etc. Hence an alternative to the above method of phase equilibria calculations is often used in which an equation of state models any vapor phase present whereas the liquid phase fugacities are found from calculation of the component activity or activity coefficient. For a liquid phase,

$$f_i^\ell = x_i \gamma_i f_i^0 \qquad (3.1.5)$$

where $f_i^{\ell}$ is the fugacity of component i in the phase, $x_i$ is the mole fraction of component i in the phase, $\gamma_i$ is the activity coefficient for component i and $f_i^0$ is called the standard state fugacity. Various expressions and simplifications are then available for calculation of $f_i^0$ and $\gamma_i$. In this research, an equation of state was used to model all phases present at equilibrium.

For a component in a mixture, the fugacity coefficient is defined as

$$\phi_i \equiv \frac{f_i}{Px_i}$$

$$\text{for } i = 1, 2, \ldots, nc \qquad (3.1.6)$$

The fugacity coefficient may be computed from

$$RT \ln \phi_i = \int_0^P [( \frac{\partial V}{\partial n_i})_{T,P,n_j} - \frac{RT}{P}]\, dP \qquad (3.1.7)$$

However, most commonly used equations of state are pressure explicit, and, after a change in variables, another more convenient expression for the fugacity coefficient of a component is given by

$$\ln \phi_i = \frac{-1}{RT}\int_\infty^V [( \frac{\partial P}{\partial n_i})_{T,P,n_j} - \frac{RT}{V}]\, dv - \ln Z \qquad (3.1.8)$$

The equation of state is differentiated and then the expression in brackets is integrated to arrive at an equation useful for calculations.

The simpler equations of state commonly used for phase equilibria calculations are two constant modfications of the van der Waals equation of state. The better known are the Redlich-Kwong [R2] equation of state and the Soave [S3] modification to the Redlich-Kwong equation of state, and the Peng-Robinson equation of state which was used in this study.

The Peng Robinson equation of state is given by

$$P = \frac{RT}{v-b} - \frac{a(T)}{v(v+b) + b(v-b)} \qquad (3.1.9)$$

Two conditions apply at the critical point. They are that

$$\left(\frac{\partial P}{\partial v}\right)_{T,cr} = 0 \qquad (3.1.10\text{-a})$$

$$\left(\frac{\partial^2 P}{\partial v^2}\right)_{T,cr} = 0 \qquad (3.1.10\text{-b})$$

$a$ and $b$ in Equation 3.1.9 can be found by application of 3.1.10 to 3.1.9 resulting in

$$a(T_c) = 0.45724\frac{R^2T_c{}^2}{P_c} \qquad (3.1.11\text{-a})$$

and

$$b(T_c) = 0.07780\frac{RT_c}{P_c} \qquad (3.1.11\text{-b})$$

where $T_c$ and $P_c$ are the critical temperature and pressure, respectively, of the substance. It may also be found that $Z_c = 0.307$. The coefficient a is made a function of temperature as follows:

$$a(T) = a(T_c)\,\alpha(T_r,\omega) \qquad (3.1.12)$$

where $\alpha(T_r,\omega)$ is a function of reduced temperature, $T/T_c$, and acentric factor, $\omega$. That is,

$$\alpha(T_r,\omega) = [1 + m(1 - T_r{}^{\frac{1}{2}})]^2 \qquad (3.1.13)$$

where

$$m = 0.37464 + 1.54226\omega - 0.26992\omega^2 \qquad (3.1.14)$$

For mixtures, the following mixing rules are often used.

$$a = \sum_{i=1}^{nc}\sum_{j=1}^{nc} x_i x_j a_{ij} \qquad (3.1.15)$$

$$b = \sum_{i=1}^{nc} x_i b_i \qquad (3.1.16)$$

where

$$a_{ij} = (1 - \delta_{ij}) a_i^{1/2} a_j^{1/2}$$

$$\text{for } i = 1, 2, \ldots, nc$$
$$\text{and } j = 1, 2, \ldots, nc \qquad (3.1.17)$$

with $\delta_{ij}$ being a binary interaction parameter. The values $a_i$ and $b_i$ are given by Equation 3.1.12 and 3.1.11 for each component in the mixture. Some researchers have used a quadratic mixing rule for the b coefficient as well as for the a coefficient [M1]. Along with such a mixing rule is used an equation similar to 3.1.17 for the $b_{ij}$ value, including a second binary interaction parameter. The value for this parameter is not the same as the value of the parameter used in Equation 3.1.17.

The compressibility factor, Z, is defined from

$$Z \equiv \frac{Pv}{RT} \qquad (3.1.18)$$

Substitution of this definition in the Peng-Robinson equation of state results in a cubic equation in Z:

$$Z^3 - (1 - B)Z^2 + (A - 3B^2 - 2B)Z$$

$$- (AB - B^2 - B^3) = 0 \qquad (3.1.19)$$

where

$$A = \frac{aP}{R^2T^2} \qquad (3.1.20)$$

and

$$B = \frac{bP}{RT} \qquad (3.1.21)$$

Using the mixing rules in Equation 3.1.15 and 3.1.16, and performing the integration in Equation 3.1.8 of Equation 3.1.9, the fugacity coefficient of a component in a mixture is given by

$$\ln \phi_i = \frac{b_i}{b}(Z - 1) - \ln (Z - B)$$

$$- \frac{A}{2\sqrt{2}B}\left( \frac{2\sum_{k=1}^{nc} x_k a_{ik}}{a} - \frac{b_i}{b}\right) \ln \left(\frac{Z + 2.414\,B}{Z - .414\,B}\right)$$

$$(3.1.22)$$

for the Peng-Robinson equation of state.

## 3.2 Material Balance Equations

Duhem's theorem is similar to the Gibbs phase rule, but lesser known. This theorem states that the equilibrium state of a closed PVT system formed from a specified initial amount of each component is completely determined by any two properties of the system provided that these two properties are independently variable at equilibrium. The variables may be either intensive or extensive. Hence pressure and temperature are two possible choices. Solving for phase compositions and amounts is called a flash calculation. Given a fixed temperature, pressure and global, or overall, composition, the composition of each phase present at equilibrium is computed as well as the relative amounts of each phase. That is, the state of the system is completely determined. For m phases present at equilibrium, the phase compositions are given as mole fractions which must satisfy the constraints

$$\sum_{i=1}^{nc} x_{ij} = 1$$

$$\text{for } j = 1, 2, \ldots, m \qquad (3.2.1)$$

where $x_{ij}$ is the mole fraction of component i in phase j. The phase compositions must satisfy material balance requirements as well. An overall mole balance of the phases must be satisfied and is given by

$$\sum_{j=1}^{m} L_j = 1 \qquad (3.2.2)$$

where $L_j$ is the ratio of moles of phase $j$ to total moles. In addition, a component balance must be satisfied

$$z_i = \sum_{j=1}^{m} x_{ij} L_j$$

$$\text{for } i = 1, 2, \ldots, nc \qquad (3.2.3)$$

where $z_i$ is the overall mole fraction of component i.

In practice, equilibrium ratios are often introduced. Here, a reference phase is chosen, and the equilibrium ratios are defined from

$$K_{ij} = \frac{x_{ir}}{x_{ij}}$$

$$\text{for } i = 1, 2, \ldots, nc \qquad (3.2.4)$$

For $j = r$ the equilibrium ratio is unity for all components. Hence it is usually taken that the $j$ index goes from $j = 1, 2, \ldots, r - 1, r + 1, \ldots, m$. When the reference phase is taken as a vapor, the equilibrium ratio is just the familiar K-value in vapor-liquid equilibrium.

From Equation 3.2.2, it is clear that only (m - 1) of the phase mole numbers are independent. For example, if three phases are present at equilibrium and one of the phases is selected as the reference phase, then

$$L_r = 1 - (L_1 + L_2) \qquad (3.2.5)$$

Using Equation 3.2.5 with 3.2.3 and the equilibrium ratios from 3.2.4, the relative amounts of each phase may be found from

$$g_j(L_1, L_2, \ldots, L_m) = \sum_{i=1}^{nc} \frac{z_i\left(\frac{1}{K_{ij}} - 1\right)}{1 + \sum\limits_{j \neq r}^{m} L_j\left(\frac{1}{K_{ij}} - 1\right)} \qquad (3.2.6)$$

There is an equation like 3.2.6 for each phase present at equilibrium not including the reference phase, that is, $j = 1, 2, \ldots, r-1, r+1, \ldots, m$. When there are three equilibrium phases for instance, Equation 3.2.6 represents two equations for the two unknowns $L_1$ and $L_2$. The equations are nonlinear, and Newton's method works well for finding the solution.

## 3.3 Conditions for a Stable Phase Equilibrium Solution

In a previous section a thermodynamic condition of equality of component chemical potentials was examined for phase equilibrium. The solution must also satisfy the material balance constraints outlined above. As pointed out by Baker, Pierce, and Luks [B1], these are necessary but not sufficient conditions for a stable phase equilibrium solution. This point can best be illustrated by the use of the Gibbs free energy of mixing function.

The difference between the value of the actual thermodynamic property and the value of the property at a reference state is called the mixing function. For the molar Gibbs free energy, $\Delta G_m$, it is given by

$$\Delta G_m = G - \sum_{i=1}^{nc} x_i G_i^0 \qquad (3.3.1)$$

where $G_i^0$ is a reference state value and $G$ is the molar Gibbs free energy of the mixture. The reference state is usually taken to be that of pure component $i$ at the same temperature and pressure as the mixture.

A phase equilibrium solution may be found geometrically using the Gibbs free energy of mixing curve. In

Figure 3.1 is shown the molar Gibbs free energy of mixing normalized by RT versus the mole fraction of component 1 in the binary mixture. Pressure is fixed at P and the temperature is also fixed at $T_1$. Since the slope of the curve is related to the chemical potential, a straight line tangent to the curve at two or more points will satisfy the criteria of equality of chemical potentials. The compositions given at the points of tangency are the compositions of the equilibrium phases. For example, in Figure 3.1, if the overall composition is chosen to be 0.2 in component 1, the mixture will split into two phases with the compositions given at A and B. Or, if the overall composition of component 1 is fixed at 0.8, the mixture will split into two phases with the compositions given at points C and D. However, if the overall composition of component 1 falls outside the intervals A to B and C to D, the mixture will remain as a single phase.

In Figure 3.2 is shown the molar Gibbs free energy of mixing curve of the same binary mixture at the same pressure but a slightly lower temperature. There is now a tangent to the curve at three points indicating that at equilibrium the mixture will split into three phases with the phase compositions given at points A, B and C when the overall composition lies in the interval AC. Again, if the

FIGURE 3.1 MOLAR GIBBS FREE ENERGY OF MIXING CURVE FOR A BINARY MIXTURE AT PRESSURE P, AND TEMPERATURE T1.

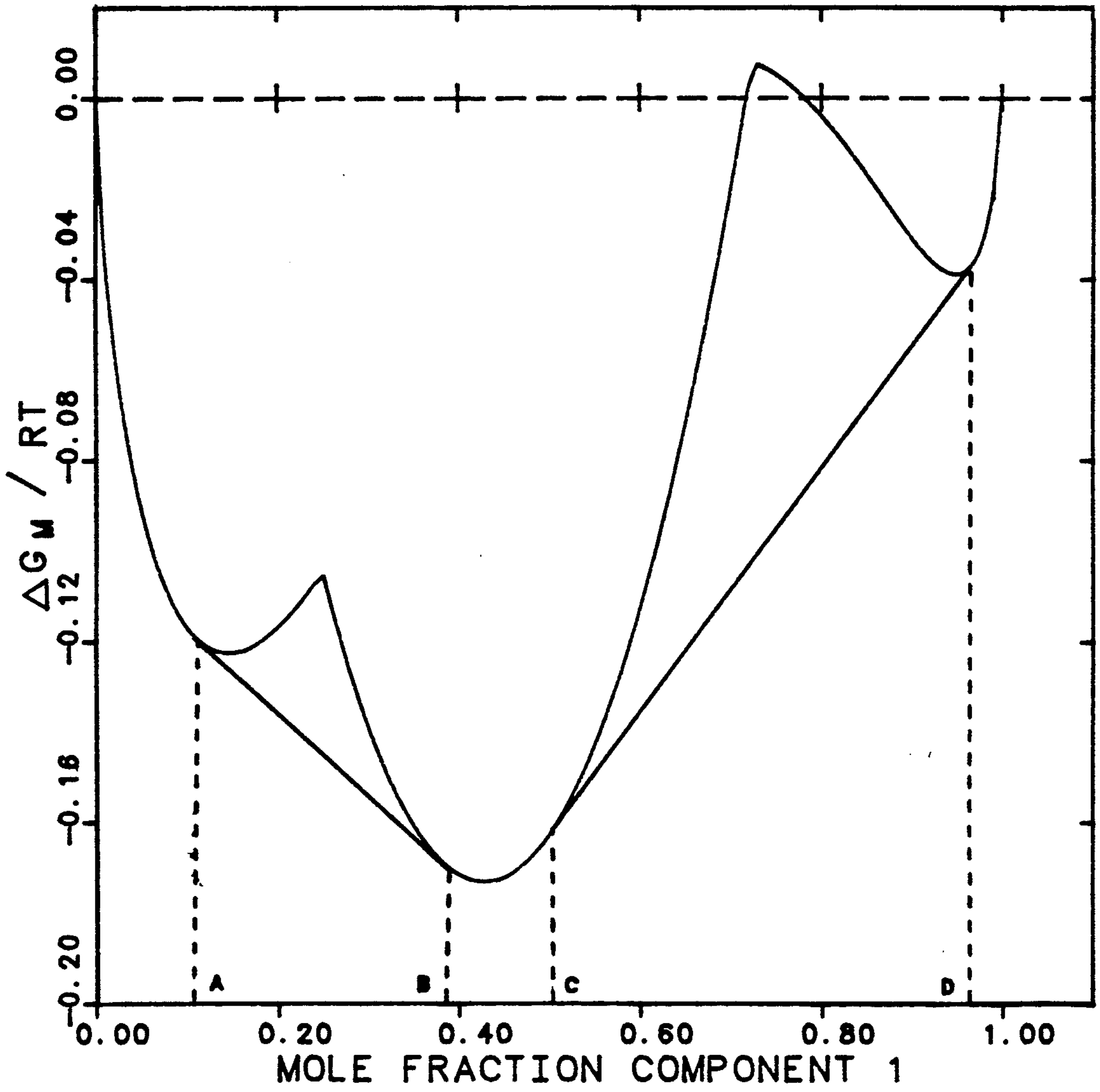
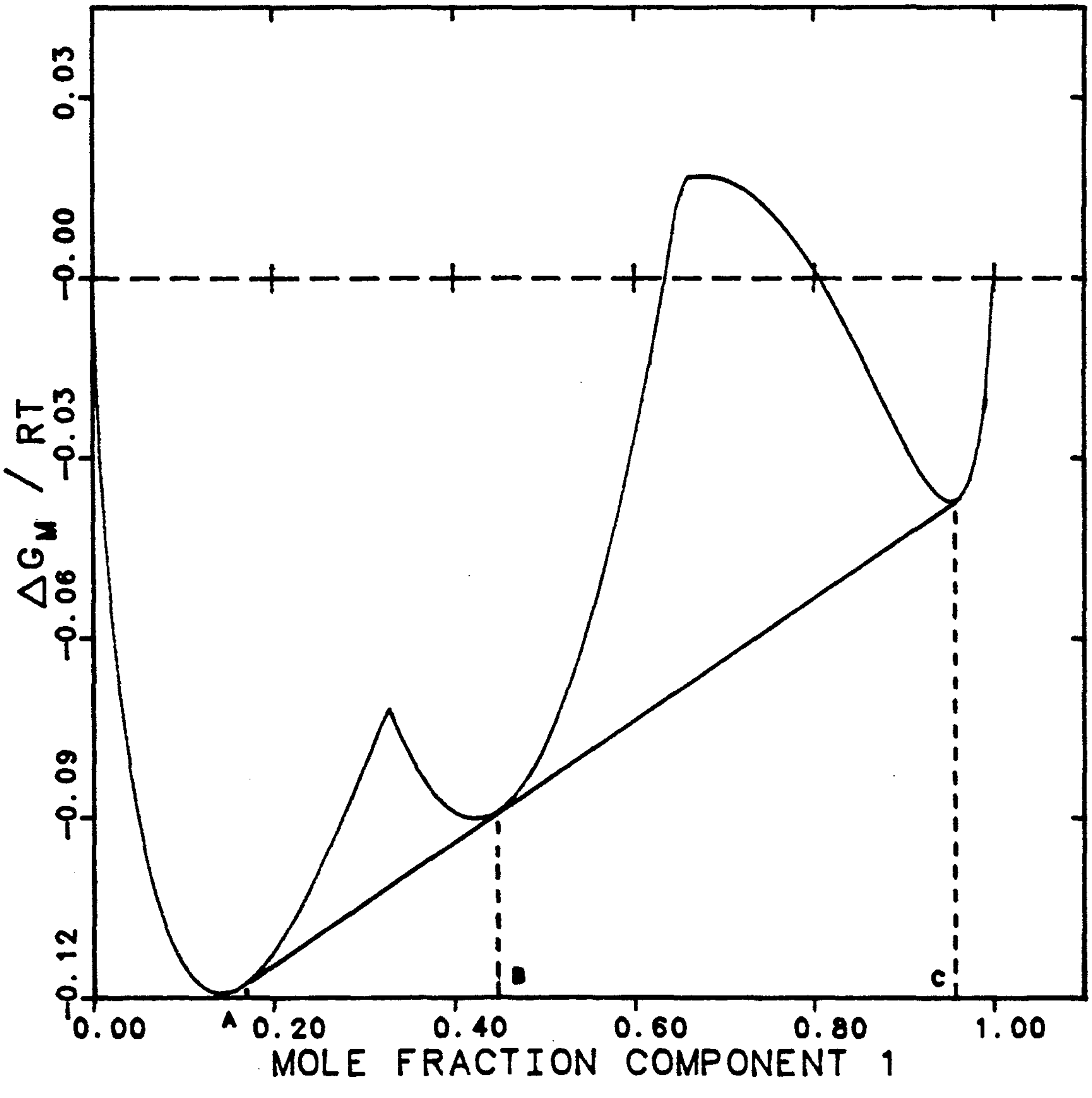


FIGURE 3.2 MOLAR GIBBS FREE ENERGY OF MIXING CURVE FOR A BINARY MIXTURE AT PRESSURE P, AND TEMPERATURE T2.



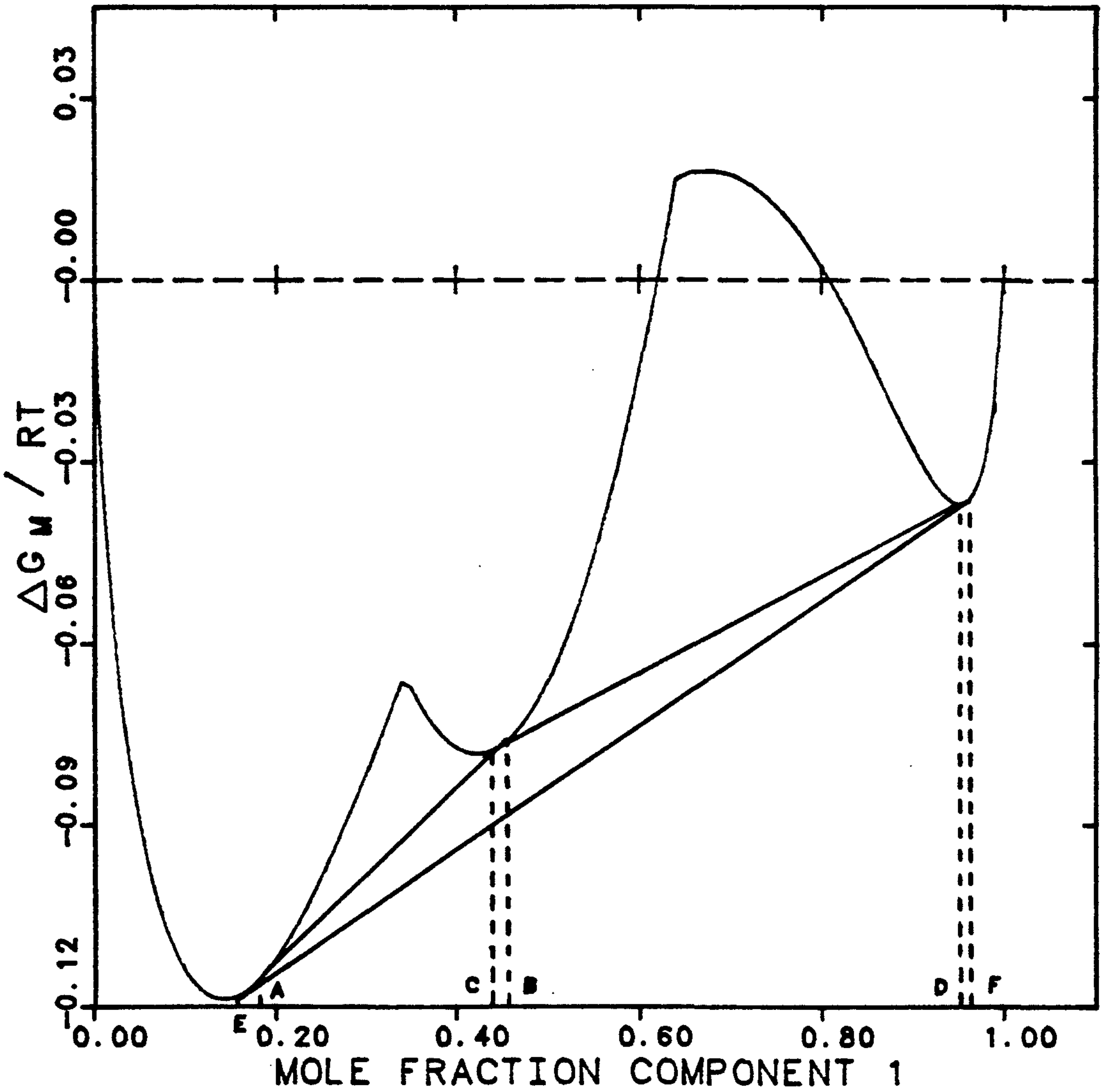
overall composition lies outside the AC interval, the mixture will be a single phase at equilibrium.

In Figure 3.3 is the binary mixture at a still lower temperature but at the same pressure as before. There are three different tangents each indicating two-phase solutions, A to B, C to D and E to F. In addition to the constraint of equal component chemical potentials for equilibrium, the equilibrium state is the state in which the Gibbs free energy is a minimum with respect to all possible changes at the given temperature and pressure. The total Gibbs free energy is the weighted sum over all phases present of the Gibbs free energy of each phase:

$$G^t = \sum_{j=1}^{m} n_j G_j \qquad (3.3.2)$$

Clearly then, the total Gibbs free energy for phases E-F is less than that of A-B for the same overall composition, and similarly for phases C-D. Thus, the stable equilibrium phase solution is given by E-F. Hence, if the overall composition falls anywhere within the interval E-F, two phases will exist at equilibrium with the phase compositions given at the points E and F rather than at A and B or C and D.

FIGURE 3.3 MOLAR GIBBS FREE ENERGY OF MIXING CURVE FOR A BINARY MIXTURE AT PRESSURE P, AND TEMPERATURE T3.



In summary then, a stable phase equilibrium solution must:

(1) satisfy the condition of equal chemical potentials for each component in each phase,

(2) satisfy material balance constraints, and

(3) satisfy the condition of minimum Gibbs free energy for all possible changes at fixed temperature and pressure.

## CHAPTER 4

## Phase Stability Analysis

Integral to the phase equilibria calculation as formulated in this research is the test of phase stability using the method developed by Michelsen [M6].

One of the difficulties in doing a flash calculation is that the number of phases present at equilibrium is not known beforehand. Thus the number and type of unknowns is not known at the outset either. The combination of phase stability analysis with a flash calculation in series can be used to help solve this problem. An important note about the phase stability anaysis is that the result does not indicate the number of phases present at equilibrium. But it does indicate whether a given mixture composition is stable as a single phase or not.

The original mixture composition is stable provided that its Gibbs free energy at fixed temperature and pressure is at a global minimum. Starting with this general thermodynamic criterion, it may be shown that an equivalent stability criterion is given by (see Appendix A):

$$\sum_{i=1}^{nc} y_i(\mu_i(y) - \mu_i(z)) \geqq 0 \qquad (4.1)$$

Here z is the composition of the original mixture and y is a trial composition. Notice that the stability analysis is done at a fixed temperature and pressure. Hence $\mu_i(y)$ and $\mu_i(z)$ indicate the functional dependence of the chemical potential on composition. It is necessary only to check the minima of the function in Equation 4.1. The stationary points are located at the zeros of the derivatives with respect to the (nc - 1) independent mole fractions. Introducing the fugacity coefficient, the above criterion may be transformed into the following set of equations:

$$\ln U_i + \ln \phi_i(y) - (\ln z_i + \ln \phi_i(z)) = 0$$

$$\text{for} \quad i = 1, 2, \ldots, nc \qquad (4.2)$$

where $\phi_i(z)$ is the fugacity coefficient of component i in the original mixture and $\phi_i(y)$ is the fugacity coefficient of component i at the location of the stationary point. The values $U_i$ are the set of independent variables. The only constraint is that these variables be non-negative. The trial composition y is a function of the independent variables and is found from

$$y_i = \frac{U_i}{\sum_{k=1}^{nc} U_k}$$

$$\text{for} \quad i = 1, 2, \ldots, nc \tag{4.3}$$

Once a solution is found, that is, the set of $U_i$ which satisfy Equation 4.2, stability of the mixture may be checked. If

$$\sum_{i=1}^{nc} U_i \leq 1 \tag{4.4}$$

then the mixture is stable as a single phase. For the sum greater than one, the mixture is unstable, and the formation of a second phase will reduce the total Gibbs free energy.

Equation 4.2 is the form of the equation which is used for calculations. The solution U may be found from several different solution methods. One technique, as Michelsen points out, is the very simple successive substitution method. The updating scheme is given by

$$\ln U_i^{n+1} = h_i - \ln \phi_i(y)^n$$

$$\text{for} \quad i = 1, 2, \ldots, nc \tag{4.5}$$

The superscripts n and n+1 indicate the iteration level. New values for y are then computed from Equation 4.3. Since the overall composition z remains the same throughout the iteration process, the terms in Equation 4.2 which depend only on z have been combined into a single variable, $h_i$, by

$$h_i = \ln z_i + \ln \phi_i(z)$$

$$\text{for} \quad i = 1, 2, \ldots, nc \tag{4.6}$$

The accelerated successive substitution method recommended by Michelsen was not implemented in this study. However, two methods higher in order than successive substituion were used--Powell's method [P3] and the secant method for solution of simultaneous non-linear equations [W2].

The set of Equations 4.2 may be rearranged

$$\ln U_i + \ln \phi_i(y) - h_i = 0$$

$$\text{for} \quad i = 1, 2, \ldots, nc \tag{4.7}$$

which is now in a form to which these methods may be applied. The secant method and Powell's method were used as implemented in the International Mathematical and Statistical Library (IMSL) [I2]. These solution techniques were available in subroutines ZSCNT and ZSPOW, accessible as online routines within the University of Texas CDC computer system.

The stability analysis as implemented uses two different solution estimates to find a single non-trivial solution. These two solution vectors are carried along through the solution procedure. Initially, equilibrium ratios are calculated using the correlation [W1]

$$K_i = \frac{T_{ci}}{P} \exp\left(5.37(1 + \omega_i)(1 - \frac{T_{ci}}{T})\right)$$

$$\text{for} \quad i = 1, 2, \ldots, nc \qquad (4.8)$$

From the equilibrium ratios, the starting solutions are set by

$$U_i{}^1 = z_i K_i \qquad (4.9)$$

and

$$U_i{}^2 = z_i / K_i \qquad 4.10)$$

corresponding to a dew point and bubble point, respectively. Here the superscripts 1 and 2 refer to the solution vectors. As a practical matter, the vectors $U^1$ and $U^2$ from Equation 4.9 and 4.10 are normalized before the stability analysis procedure is begun. Also, the starting solutions are adjusted so that the dominant component in the starting vectors are not both greater or both less than that component in the original mixure.

Usually, one of the two solution vectors will converge to a trivial solution, that is,

$$U_i = z_i$$

$$\text{for } i = 1, 2, \ldots, nc \qquad (4.11)$$

while the second solution vector converges to a non-trivial solution. By using two different solution vectors, with the modification indicated previously, a non-trivial solution is found, provided one exists.

# CHAPTER 5

## Solution Algorithm

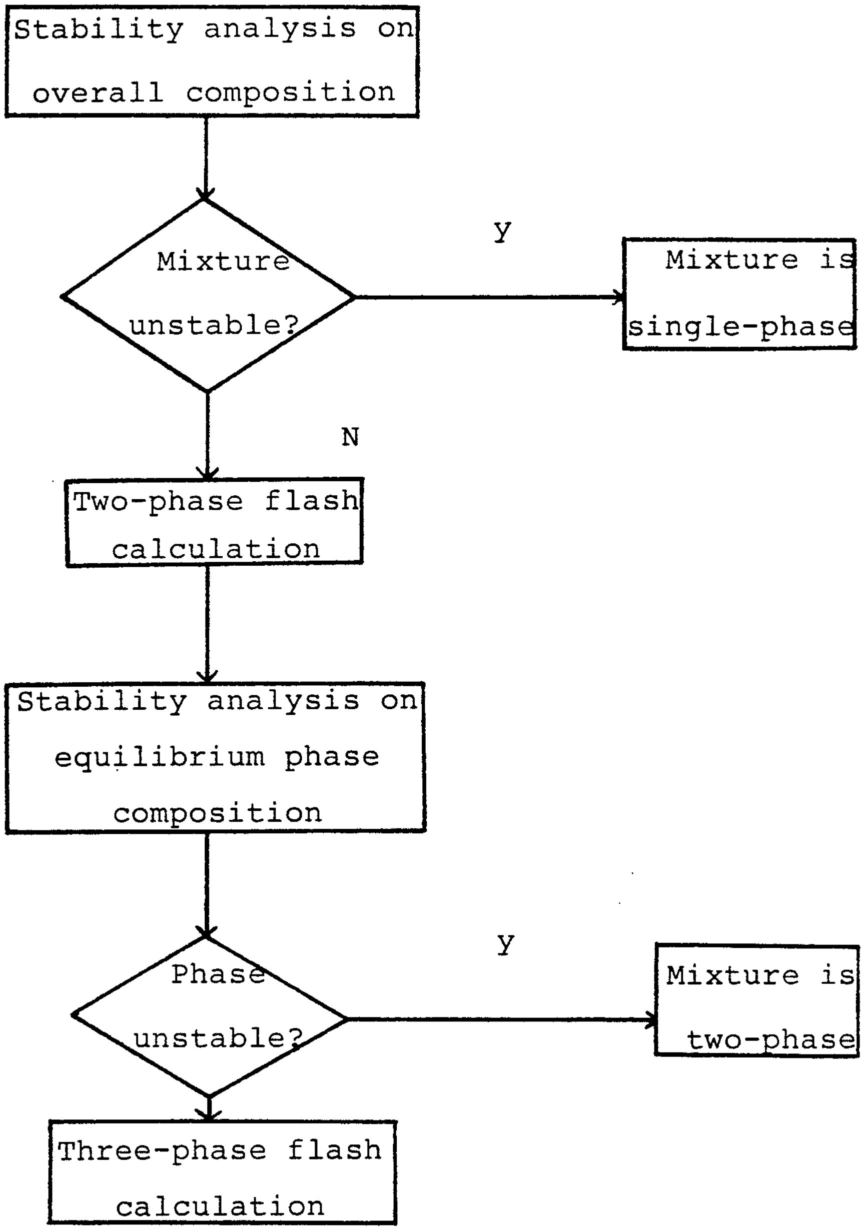
In this chapter the algorithm for multi-component, multi-phase flash calculations as used in this study will be described. In the first section, a general overview of the logic will be presented. More detail is then given in following sections about specific portions of the algorithm.

### 5.1 Overall Logic of Algorithm

The purpose of the algorithm is to predict the number of phases present at equilibrium, the composition of each of the equilibrium phases, and the relative amounts of the phases given a fixed temperature, pressure, and overall composition. The algorithm is arranged in a step-wise fashion. Figure 5.1 presents a flow-chart of the major divisions or steps that the algorithm contains.

Initially, the phase stability analysis is tested on the overall composition which checks for the possibility of two phases. If the test indicates stability, then the mixture is taken to be single-phase and no further calculations are necessary.

Figure 5.1 Overall procedure for multi-phase flash algorithm.



If the test indicates an instability, initial phase composition estimates are set for each of the two phases and a two-phase flash calculation is done.

The third step is to check the stability of one of the two phases from the flash calculation. This check of phase stability is similar to the initial phase stability analysis except that the analysis is done using a phase composition rather than the overall composition.

If the results indicate that the phase is stable, the mixture is assumed to be two-phase and no further calculation is necessary. If results indicate an instability, then a three-phase flash calculation is carried out.

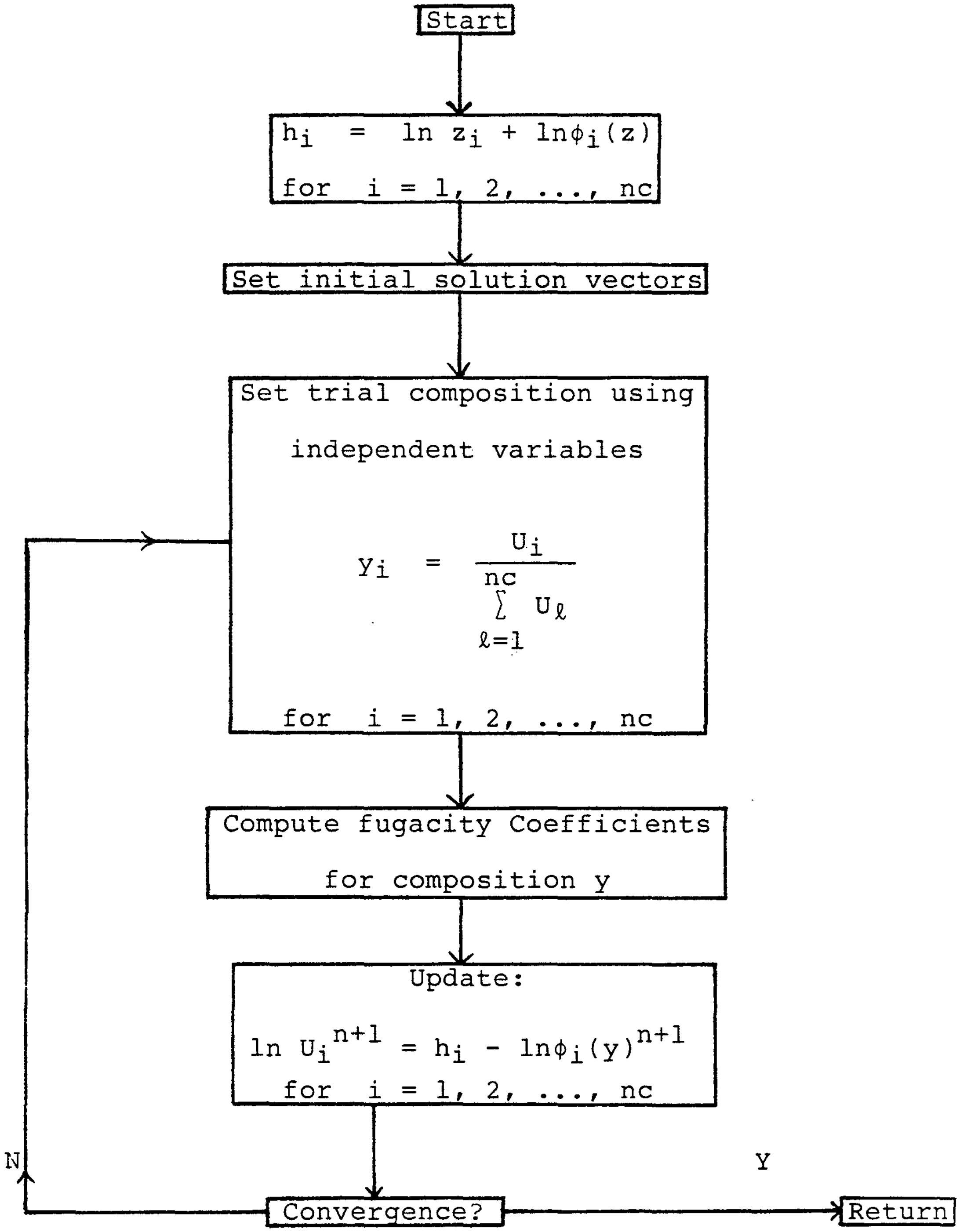
Thus the algorithm proceeds in a step-wise fashion in which phase stability is analyzed followed by the addition of a phase, if necessary, up to a maximum of three coexisting phases.

## 5.2 Initial Two-Phase Stability Analysis

The initial phase stability analysis is the first step in the flash calculation. The stability is checked using the overall composition which will render a result indicating either a single phase mixture at equilibrium or that more than one phase should exist.

Figure 5.2 shows a flow chart for the solution of the stability analysis criteria using the successive

Figure 5.2 Schematic for successive substitution method of solution of phase stability analysis.



substitution method. As indicated in Chapter 4, the stability analysis makes use of two solution vectors. Even though the starting values of each vector are set at the same time, the individual solutions may be found either in parallel or in series.

The trial composition is found by normalizing the values of the independent variables as in Equation 4.3. The fugacity coefficients are calculated using the equation of state and updated by Equation 4.5. Iteration proceeds until

$$\sum_{i=1}^{nc} (\ln U_i + \ln \phi_i(y) - h_i)^2 \leqq \varepsilon \qquad (5.2.1)$$

where $\varepsilon$ is a small value, for example $10^{-10}$ or $10^{-12}$.

When calculating the fugacity coefficient from the equation of state, Equation 3.1.22, the compressibility factor must first be calculated. As shown in Equation 3.1.19, the Peng-Robinson equation of state is cubic in compressibility factor. In this study, the roots were found using a direct method. Since the equation is cubic, the roots may all be real or it may be that only one real root exists with the other two roots being complex. When three real roots exist, the compressibility factor is chosen so that the Gibbs free energy is minimized. The compressibility factor is commonly chosen such that the smallest, non-negative real root corresponds to a liquid phase while the largest real root corresponds to a vapor phase. Since knowledge of the type of phase is not always available, any ambiguities are avoided by using the Gibbs free energy for root selection.

In addition to the successive substitution method, the secant method and Powell's method were also implemented for solution of the phase stability criteria. Both methods were accessible from the IMSL library routines. To use either method, a subroutine must be provided by the user which calculates the set of equations in a residual form, that is, in the form of Equation 4.7. After the initial solution estimates are set, the library routines are called and the converged solutions are returned.

Once the solutions are found, the results must be interpreted. A trivial solution is a solution which has the same composition as the overall mixture. This possibility must be checked. The other possibility is a solution which has a composition differing from the overall mixture composition. When this occurs, the sum of the independent variables is found. If the sum is less than or equal to one, the mixture is stable as a single phase. A sum greater than one indicates that at least two phases should exist at equilibrium. In general, one solution results in a non-trivial solution while the other solution is trivial, but in some cases, neither solution is trivial. When both solutions are trivial, it is assumed that the mixture is stable as a single phase.

## 5.3 Two-Phase Flash Calculation

If the results from the stability analysis indicate that the mixture is unstable as a single phase, a flash calculation constrained to two equilibrium phases is the next step.

### 5.3.1 Generation of Starting Values

An important consideration of the flash calculation is the initial values of the phase compositions. Since the calculation is an iterative process, poor starting values may lead to convergence problems. A common starting point used in flash calculations is to utilize the correlation of Equation 4.8 to compute an equilibrium ratio for each component present in the mixture. Using this set of equilibrium ratios, the phase distribution, that is, the relative amounts of each phase, may be found. With this result and the equilibrium ratios, the phase compositions can be set and iteration may begin. However, as an alternative, the results from the previous stability analysis can be used to generate starting phase compositions.

When an instability of the original mixture is indicated, two situations involving the types of solutions will arise. It may be that neither solution is trivial or it may be that one solution is trivial and the other is not. To generate the starting values, these two cases are dealt with differently.

For the case in which both solutions are non-trivial, a check must be made so that they do not represent the same solution. If they do, one of the solutions is treated as a trivial solution and considered with the second case. If the two solutions are different, their values are normalized and then used as the starting phase compositions. The phase distribution is found by solution of the following equation for any component:

$$L_1 = \frac{z_i - x_{i2}}{x_{i1} - x_{i2}} \qquad (5.3.1)$$

Since $L_2 = 1 - L_1$, a complete solution estimate is set and iteration may begin.

The second case has a trivial solution and a non-trivial solution. Here, the non-trivial solution is normalized and used as the starting composition of one phase. The composition of the second phase is easily calculated once the phase distribution is set.

Consider a mixture which has two equilibrium phases for a specified temperature and pressure. If the overall composition is close to the composition of one of the equilibrium phases, then a non-trivial solution, when normalized, will give a composition which is close to the composition of the other equilibrium phase. This is the reason that the non-trivial solutions were used for starting values as described above. However, it is not known beforehand how close the overall composition is to a phase composition. But $L_1$ or $L_2$ must be set.

It was found that as the overall composition moves from one phase composition toward the other, the sum of the independent variables, Equation 4.4, gets larger and then decreases as the other phase composition is approached. If the phase for which the composition has been set is called phase 1, and the other phase 2, the phase mole number of phase 2, $L_2$, is set equal to the inverse of the sum. Since the choice of $L_2$ is somewhat arbitrary, this method has the benefit of being very simple. Also, since the sum is clearly greater than one, the inverse is always between zero and one, which is the interval in which $L_2$ must lie. Although there is no strict relationship, the phase distribution is loosely connected with the stability analysis.

Finally, once $L_1$ and $L_2$ are set, the composition of the second phase may be calculated and iteration begun.

### 5.3.2 Solution Methods of Flash Equations

In this study, three methods were used for solution of the flash equations. They are the accelerated successive substitution method, Powell's method, and the secant method for simultaneous non-linear equations.

The condition for phase equilibrium may be written as

$$\ln (x_{il}\phi_{il}) - \ln (x_{ir}\phi_{ir}) = 0 \qquad \text{for } i = 1, 2, \ldots, nc \qquad (5.3.2)$$

For the two-phase case, there are nc equations of this type which may be solved for nc independent variables. The accelerated successive substitution (ACSS) method will be discussed separately from Powell's method and the secant method, the two of which are accessed from the IMSL Library.

### The Accelerated Successive Substitution Method

The ACSS method is a combination of the pure successive substitution method presented by Nghiem and Aziz [N2] and an acceleration scheme developed by Mehra, Heidemann, and Aziz [M2] for phase behavior calculations. A reference phase is selected and the set of equilibrium ratios are computed. The equilibrium phase compositions are treated as dependent variables. Then, Equation 5.3.2 is solved for the equilibrium ratios, and with Equations 3.2.4 and 3.2.6, the dependent variables may be computed.

Figure 5.3 illustrates the iterative procedure for the ACSS method. The first step is to compute the phase distribution for the given set of equilibrium ratios. Applying Equation 3.2.6 to the two-phase case, the phase distribution is found by solving for $L_1$:

$$g(L_1) = \sum_{i=1}^{nc} \frac{z_i(\frac{1}{K_i} - 1)}{1 + L_1(\frac{1}{K_i} - 1)} = 0 \qquad (5.3.3)$$

This equation is non-linear for $L_1$ and may be solved using Newton's method.

Once the phase distribution is set, the phase compositions are calculated from

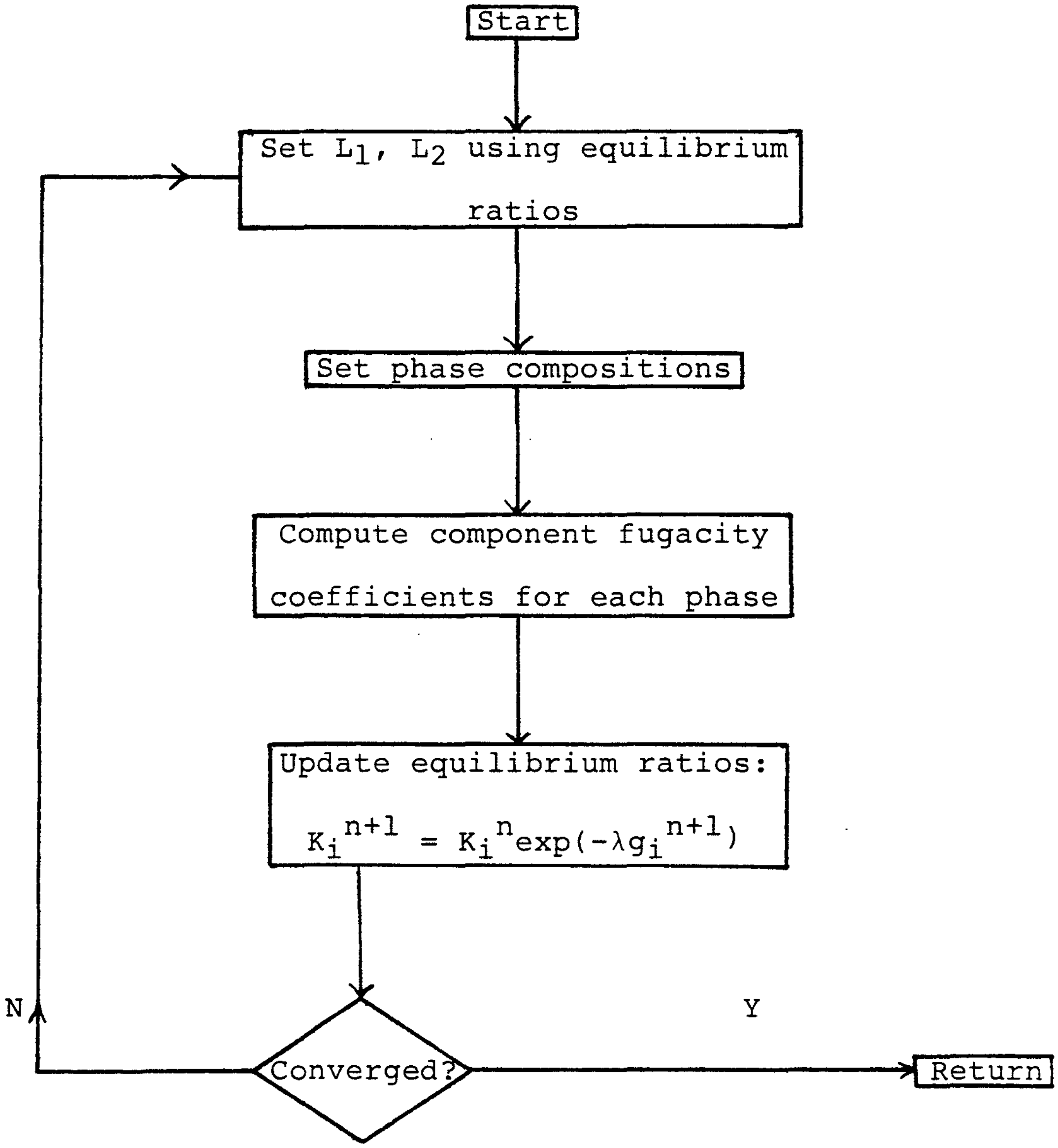
$$x_{ir} = \frac{z_i}{1 + L_1(\frac{1}{K_i} - 1)} \qquad (5.3.4)$$

$$x_{i1} = x_{ir}/K_i \qquad (5.3.5)$$

$$\text{for} \quad i = 1, 2, \ldots, nc$$

After the phase compositions have been computed, the component fugacity coefficients are calculated for each

Figure 5.3 Flowchart for accelerated successive substitution method



phase using the equation of state. The equilibrium ratios are then updated.

For pure successive substitution, the updating scheme is given by

$$K_i^{n+1} = K_i^n \left(\frac{f_{il}}{f_{ir}}\right)$$

$$\text{for} \quad i = 1, 2, \ldots, nc \qquad (5.3.6)$$

In terms of the fugacity coefficients,

$$K_i^{n+1} = \frac{\phi_{il}}{\phi_{ir}}$$

$$\text{for} \quad i = 1, 2, \ldots, nc \qquad (5.3.7)$$

The acceleration scheme is given by

$$K_i^{n+1} = K_i^n \exp\left(- \lambda^{n+1} g_i^{n+1}\right)$$

$$\text{for} \quad i = 1, 2, \ldots, nc \qquad (5.3.8)$$

where $\lambda$ is an acceleration parameter, found from

$$\lambda^{n+1} = \left| \frac{\lambda^{n} \sum_{i=1}^{nc} [g_i^{n}]^2}{\sum_{i=1}^{nc} [g_i^{n+1} g_i^{n}] - \sum_{i=1}^{nc} [g_i^{n}]^2} \right| \qquad (5.3.9)$$

with $\lambda^1 = 1$ where

$$g_i = \ln \left(\frac{f_{ir}}{f_{il}}\right) \qquad (5.3.10)$$

For the first iteration, n = 1 and the equilibrium ratios are updated by successive substitution.

After the equilibrium ratios are updated, convergence is checked. In this study, the following criteria were used [T1]:

$$\sum_{i=1}^{nc} \frac{(K_i^{n+1} - K_i^{n})^2}{K_i^{n+1} K_i^{n}} \leqq \varepsilon \qquad (5.3.11)$$

where $\varepsilon$ is a small value, usually between $10^{-15}$ and $10^{-20}$. If convergence has not been achieved, the phase distribution is set using the updated equilibrium ratios and iteration is continued. After the convergence criteria are met, the algorithm proceeds with another phase stability analysis.

## Powell's Method

As implemented, the component mole numbers were selected as the independent variables for which Equation 5.3.2 is solved, that is, the set of variables $n_{ij}$ where

$$n_{ij} = L_j x_{ij} \qquad \text{for } i = 1, 2, \ldots, nc \qquad (5.3.12)$$

Here j is a specified phase and is chosen to be that phase which has the smallest number of moles present. To use the IMSL routine, a subroutine which computes the functions in the form of Equation 5.3.2 must be provided. The mole numbers are used along with the overall composition to set the phase mole fractions before the fugacity coefficients may be calculated. Both the Powell's method and the secant method use the same subroutines.

## 5.4 Stability Analysis Test for an Additional Phase

The purpose of the stability analysis after the two-phase flash is to test for an instability of the equilibrium phases just calculated. If the results indicate that the phases are unstable, a third phase will be added and another flash calculation will be done. In this way, the algorithm checks for possible three-phase equilibrium.

The procedure of the stability analysis at this point in the algorithm is very similar to that of the initial stability analysis. There are three major differences. First, the composition tested is a phase composition calculated by the two-phase flash as opposed to the overall composition. The second difference is that there are two possible trivial solutions, which are the equilibrium phase compositions. Thirdly, four solution vectors are used instead of two.

One solution estimate is set to be nearly pure in the heaviest hydrocarbon component present, and a second solution estimate is set to be nearly pure in the lightest hydrocarbon component, or a non-hydrocarbon if present in the mixture. The third solution is set to be an average between the two phase compositions from the flash calculation. The fourth and final solution estimate is found from

$$U_i = \exp(h_i) \qquad \text{for } i = 1, 2, \ldots, nc \tag{5.4.1}$$

where $h_i$ is a function of the composition being tested.

The meaning of the final solutions is the same as that in the initial phase stability analysis. If all four solutions indicate that the phase is stable, then the mixture is considered to have only two phases present at equilibrium and no further calculations are necessary. However, if any of the stability results indicate the phase is unstable, another phase is added and a three-phase flash calculation is started.

## 5.5 Three-Phase Flash Calculation

As noted before, if the phase equilibrium solution from the two-phase flash calculation is found to be unstable, an additional phase is assumed to exist at equilibrium. A three-phase flash calculation is done to find the compositions and relative amounts of the phases. This is the last step in the algorithm.

### 5.5.1 Starting Values

Unlike the two-phase vapor-liquid case, there is no empirical relationship with which either equilibrium ratios or phase compositions may be estimated for three coexisting phases. As in the two-phase flash, the results of the stability analysis may be used to formulate initial phase compositions. The phase composition estimates which may be used are the two phase compositions from the preceding flash calculation and the estimate found from the phase stability analysis. However, these values do not satisfy the material balance constraints necessary for the three-phase flash calculation.

The method developed in this study uses these three composition vectors indirectly to set the starting values. The first step is normalize the solution from the stability analysis to a mole fraction form. The three sets of values are used as ratios to find the component mole numbers for each phase. If the three composition vectors above are called $w_{ij}$, for i = 1, 2, ..., nc and j = 1, 2, 3, the mole numbers are found from

$$n_{ij} = \frac{w_{ij} z_i}{\sum_{k=1}^{nc} w_{ik}}$$

$$\text{for} \quad i = 1, 2, \ldots, nc$$

$$\text{and} \quad j = 1, 2, 3 \qquad (5.5.1)$$

These component mole numbers are added for each phase to set the initial phase distribution. This method is very simple, but yields initial estimates which satisfy the material balance constraints without making ambiguous restrictions on the compositions and phase distributions.

### 5.5.2 Three-Phase Flash

For the three-phase case, the condition for phase equilibrium, like Equation 5.3.2, is given by

$$\ln (x_{ij}\phi_{ij}) - \ln (x_{ir}\phi_{ir}) = 0$$

$$\text{for } i = 1, 2, \ldots, nc$$

$$\text{and } j = 1, 2 \qquad (5.5.2)$$

There are (2 nc) equations of this form which are solved for (2 nc) independent variables.

The methods of solution for the three-phase flash are the same as for the two-phase flash. The equations used are extensions of the two-phase flash equations described earlier.

Like the two-phase flash, for the accelerated successive substitution method, one phase is selected as a reference phase. From this, two sets of equilibrium ratios may be defined. Equation 3.2.6 in the three-phase case forms a set of two simultaneous equations in the two unknowns $L_1$ and $L_2$. Again, this system of equations may be solved using Newton's method.

The equilibrium ratios are updated by

$$K_{ij}^{n+1} = K_{ij}^{n} \exp(-\lambda^{n+1} g_{ij}^{n+1})$$

$$\text{for } i = 1, 2, \ldots, nc$$

$$\text{and } j = 1, 2 \qquad (5.5.3)$$

where the acceleration factor is calculated from

$$\lambda^{n+1} = \left| \frac{\lambda^{n} \sum_{j=1}^{2} \sum_{i=1}^{nc} [g_{ij}{}^{n}]^{2}}{\sum_{j=1}^{2} \sum_{i=1}^{nc} [g_{ij}{}^{n+1} g_{ij}{}^{n}] - \sum_{j=1}^{2} \sum_{i=1}^{nc} [g_{ij}{}^{n}]^{2}} \right| \tag{5.5.4}$$

where

$$g_{ij} = \ln\left(\frac{f_{ir}}{f_{ij}}\right)$$

$$\text{for} \quad i = 1, 2, \ldots, nc$$

$$\text{and} \quad j = 1,2 \tag{5.5.5}$$

Again, for the first iteration, $n = 1$ and $\lambda^1 = 1$, which is a pure successive substitution step.

Iteration is continued until the following condition is satisfied:

$$\sum_{j=1}^{2} \sum_{i=1}^{nc} \frac{(K_{ij}{}^{n+1} - K_{ij}{}^{n})^2}{K_{ij}{}^{n+1} K_{ij}{}^{n}} \leqq \varepsilon \tag{5.5.6}$$

which is an extension of the two-phase case given in Equation 5.3.11.

For Powell's method and the secant method, the independent variables consist of the component mole numbers of two of the three phases. The solution procedure is nearly identical to that of the two-phase case with the exception of having an additional nc independent variables.

# CHAPTER 6

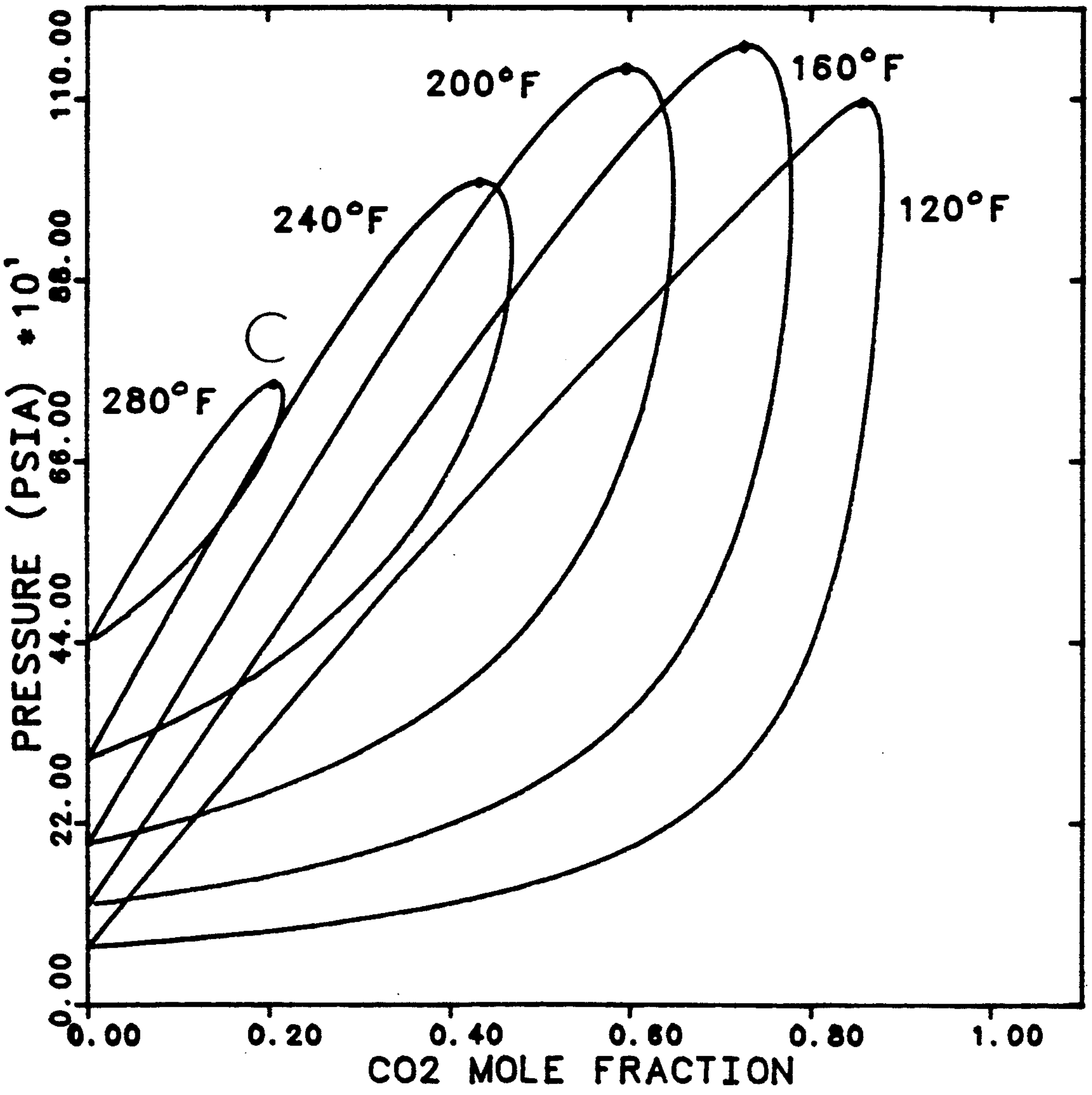
# Results

In this chapter, several examples of phase equilibria calculations are presented. Mixtures showing two- and three-phase behavior are examined. In addition, a comparison is made of the various solution methods for the phase stability analysis equations under various conditions. The Peng-Robinson equation of state has been used for all phase equilibria calculations presented here.

## 6.1 $CO_2$-n-Butane Binary Mixture

In Figure 6.1 is shown a pressure-composition diagram for the binary mixture of carbon dioxide and normal butane at several temperatures. This and subsequent calculations were made using Powell's method for solution of the phase stability equations and the accelerated successive substitution method for solution of the flash equations. The tolerance used for the phase stability analysis was set so that the computed root was accurate to eight digits according to IMSL documentation [I2]. For the ACSS method the value of $\varepsilon$ in Equation 5.3.11 was set to $10^{-18}$ for the three-phase calculation where the convergence criterion is

FIGURE 6.1 COMPUTED LIQUID-VAPOR PHASE ENVELOPE FOR THE CO2 AND N-BUTANE BINARY MIXTURE AT SEVERAL TEMPERATURES.



given by Equation 5.5.6. To find the phase distribution, Newton's method was used. In this calculation, iteration was stopped when

$$|g(L_1)| \leqq 10^{-10} \qquad (6.1.1)$$

where $g(L_1)$ is given by Equation 5.3.3 for the two-phase flash and for the three-phase flash,

$$(g_1(L_1,L_2)^2 + g_2(L_1,L_2)^2)^{\frac{1}{2}} \leqq 10^{-10} \qquad (6.1.2)$$

where $g_j(L_1,L_2)$ are given by Equation 3.2.6.

The critical constants and acentric factors used as input for the equation of state are listed in Table 6.1. Values for all components used in the example calculations were taken from this table unless indicated otherwise for specific examples. Listed in Table 6.2 are binary interaction coefficient values that were used in the phase equilibria calculations. The primary literature source was Nagy and Shirkovskiy [N1], but some values were taken from Lawal [L1] as well as Risnes [R4].

There is a great deal of variation in the values given for binary interaction coefficients in the literature. Nagy, for example, lists the $CO_2$-$CH_4$ binary interaction coefficient as 0.105, Lawal gives a value of 0.097, Risnes gives a value of 0.100, and Mehra [M1] lists the value at

Table 6.1

Component critical constants and acentric factors used in example calculations.

| | Critical | Critical | |
|---|---|---|---|
| Component | Temperature (R) | Pressure (psia) | Acentric Factor |
| $CO_2$ | 547.58 | 1071.40 | .225 |
| $CH_4$ | 343.04 | 668.05 | .010 |
| $C_2H_6$ | 549.72 | 708.35 | .105 |
| $C_3H_8$ | 665.82 | 617.23 | .152 |
| i-$C_4H_{10}$ | 734.58 | 529.06 | .192 |
| n-$C_4H_{10}$ | 765.31 | 550.66 | .201 |
| i-$C_5H_{12}$ | 829.80 | 483.50 | .206 |
| n-$C_5H_{12}$ | 845.64 | 489.38 | .252 |
| n-$C_6H_{14}$ | 914.22 | 439.41 | .290 |
| n-$C_{10}H_{22}$ | 1114.20 | 305.68 | .586 |
| n-$C_{16}H_{34}$ | 1290.60 | 205.81 | .704 |
| $C_{7+}$ | 1395.90 | 232.20 | .619 |

## Table 6.2

Binary interaction parameters used in example calculations.

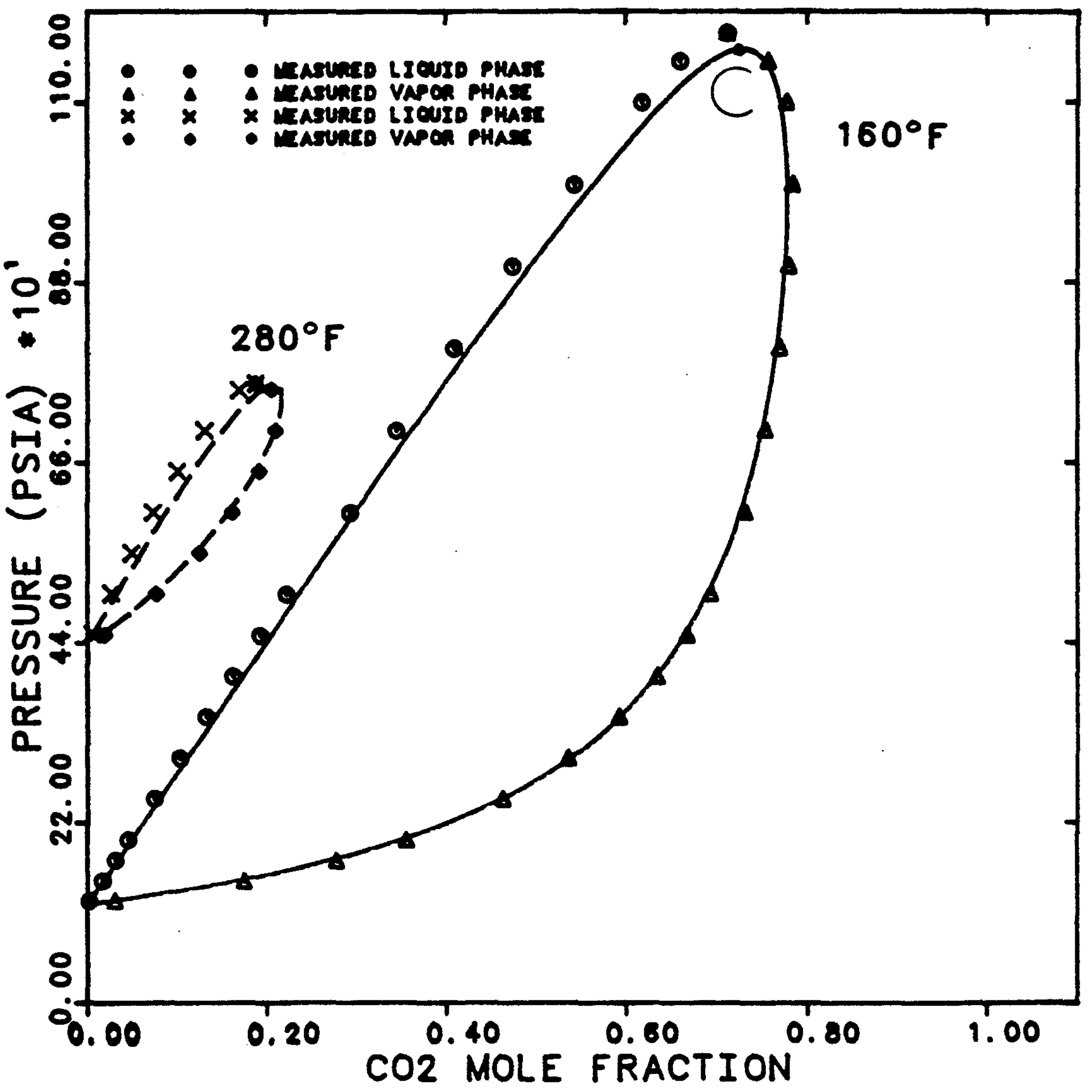
| | $CO_2$ | $CH_4$ | $C_2H_6$ | $C_3H_8$ | i-$C_4$ | n-$C_4$ | i-$C_5$ | n-$C_5$ | n-$C_6$ |
|---|---|---|---|---|---|---|---|---|---|
| $CH_4$ | .100 | 0 | | | | | | | |
| $C_2H_6$ | .102 | .003 | 0 | | | | | | |
| $C_3H_8$ | .107 | .010 | .002 | 0 | | | | | |
| i-$C_4$ | .115 | .019 | .006 | .002 | 0 | | | | |
| n-$C_4$ | .115 | .019 | .006 | .002 | 0 | 0 | | | |
| i-$C_5$ | .118 | .028 | .011 | .005 | .001 | .001 | 0 | | |
| n-$C_5$ | .118 | .028 | .011 | .005 | .001 | .001 | 0 | 0 | |
| n-$C_6$ | .123 | .037 | .017 | .009 | .005 | .004 | .001 | .001 | 0 |
| n-$C_{10}$ | .115 | | | | | | | | |
| n-$C_{16}$ | .125 | 0 | | .053 | | | | | |
| $C_{7+}$ | .250 | .129 | .073 | .054 | .044 | .042 | .035 | .034 | .028 |

0.103. The range for the $CO_2$-n-decane binary interaction coefficient is from 0.110 (Mehra), to 0.190 (Lawal). For $CO_2$-n-hexadecane, Mehra gives a value of 0.09, Risnes lists a value of 0.125, and Lawal gives a value of 0.248. Although these parameters are usually found using correlations fitted from experimental data, a large variation in reported values still remains. Adjustment of these values for use in calculations is justified based on this variation, and in many cases, large adjustments can be made while still remaining within the range of reported values.

In Figure 6.1, the mole fraction of $CO_2$ in the phase is plotted along the x-axis. The highest temperature shown is 280°F. The mixture shows typical liquid-vapor behavior. All temperatures are higher than the critical temperature of carbon dioxide; therefore the liquid and vapor phase binodals meet at a critical point. Tie lines which connect equilibrium phase compositions are horizontal lines since there are only two components in the mixture. As the temperature increases, the size of the two-phase region decreases. Once the temperature is above about 305°F, which is the critical temperature of pure normal-butane, all mixtures will be single phase.

Figure 6.2 shows a comparison between the calculated equilibrium phase compositions and experimental values. The experimental data was taken from Olds, Reamer, Sage,

FIGURE 6.2 PRESSURE VERSUS COMPOSITION DIAGRAM FOR THE CO2 AND N-BUTANE MIXTURE. COMPARISON OF CALCULATED AND MEASURED PHASE COMPOSITION AT TWO TEMPERATURES.



and Lacey [O1]. Two temperatures are shown, namely 280°F and 160°F. Although no adjustment of input data was made, a good comparison with the experimental data can be seen. The vapor phase compositions are modeled slightly better than are the liquid phase compositions at both temperatures. This is more pronounced near the critical point.

Figure 6.3 shows calculated phase compositions with the experimental data of Poettmann and Katz [P2] at 200°F and 120°F. Experimental data are quite sparse. Yet calculated values compare well with the experimental data.

### 6.2 $CO_2$-n-Decane Binary Mixture

A molar Gibbs free energy of mixing curve for the $CO_2$-n-decane binary mixture at 100°F and 1300 psia is shown in Figure 6.4. Because the mixing curve is so flat near the two-phase region, small variations in the state of the system will produce large changes in phase behavior. This was apparent in calculations.

In addition to forming liquid and vapor phases, $CO_2$ and n-decane mixtures show liquid-liquid-vapor equilibrium as illustrated in Figure 2.1. An example of liquid-liquid phase equilibria on a pressure-composition diagram is shown in Figure 6.5 for this binary mixture. The temperature is fixed at -13°F and -31°F. The value of 0.115 was used for the binary interaction coefficient

FIGURE 6.3 PRESSURE VERSUS COMPOSITION DIAGRAM FOR THE CO2 AND N-BUTANE MIXTURE. COMPARISON OF CALCULATED AND MEASURED PHASE COMPOSITION AT TWO TEMPERATURES.

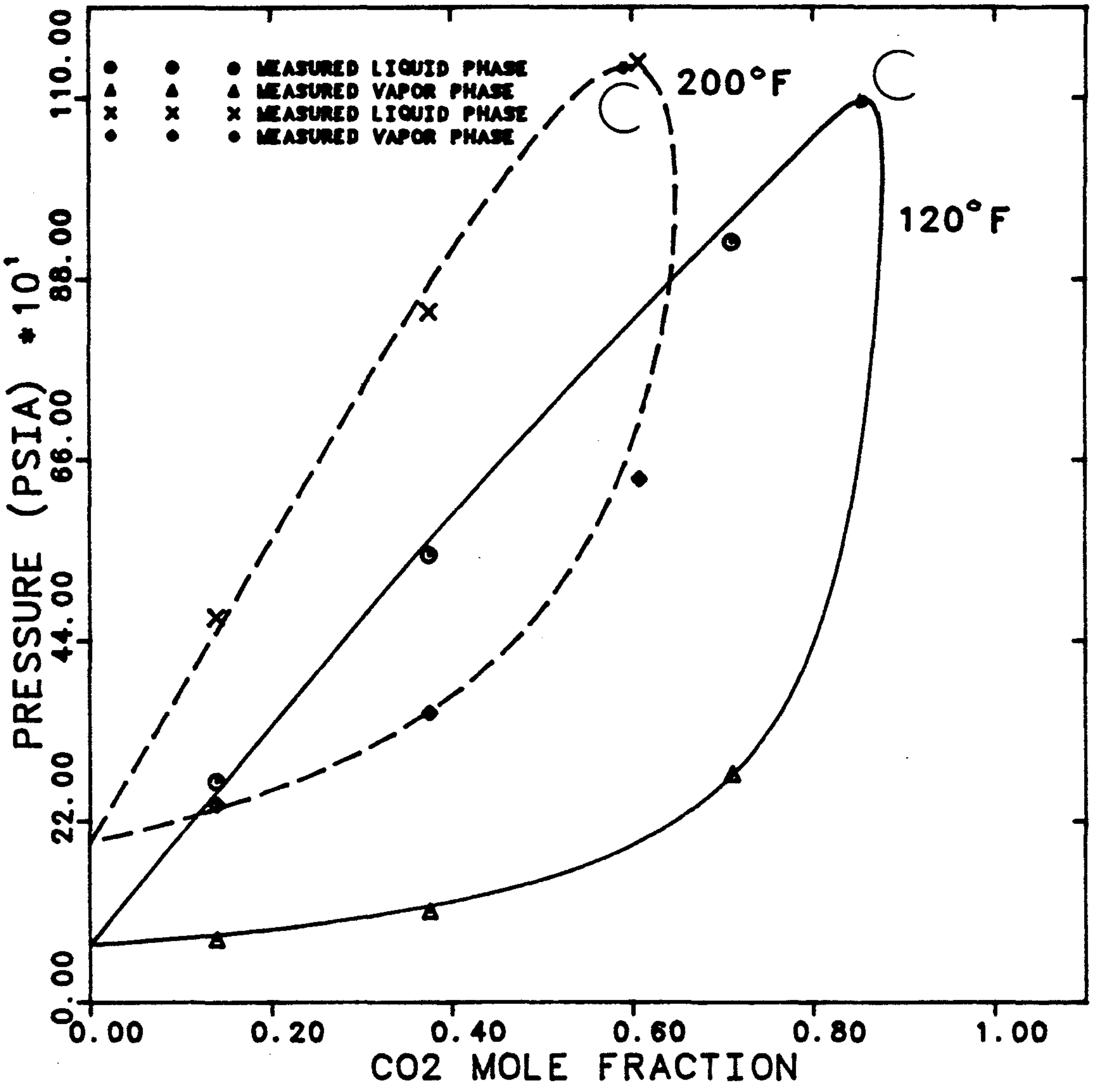


FIGURE 6.4 MOLAR GIBBS FREE ENERGY OF MIXING CURVE FOR A BINARY MIXTURE OF CO2 AND N-DECANE AT 100°F AND 1300 PSIA.

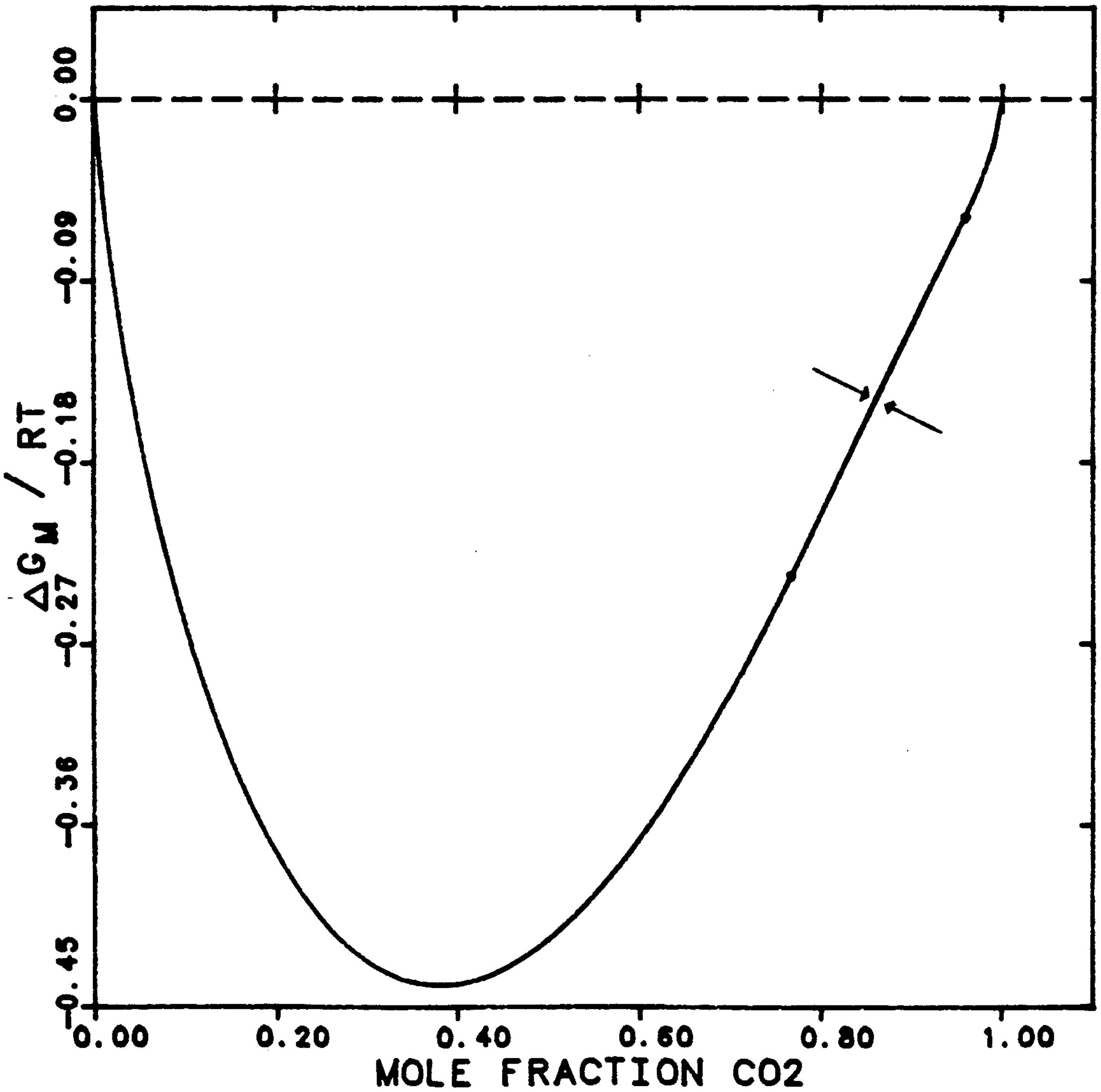
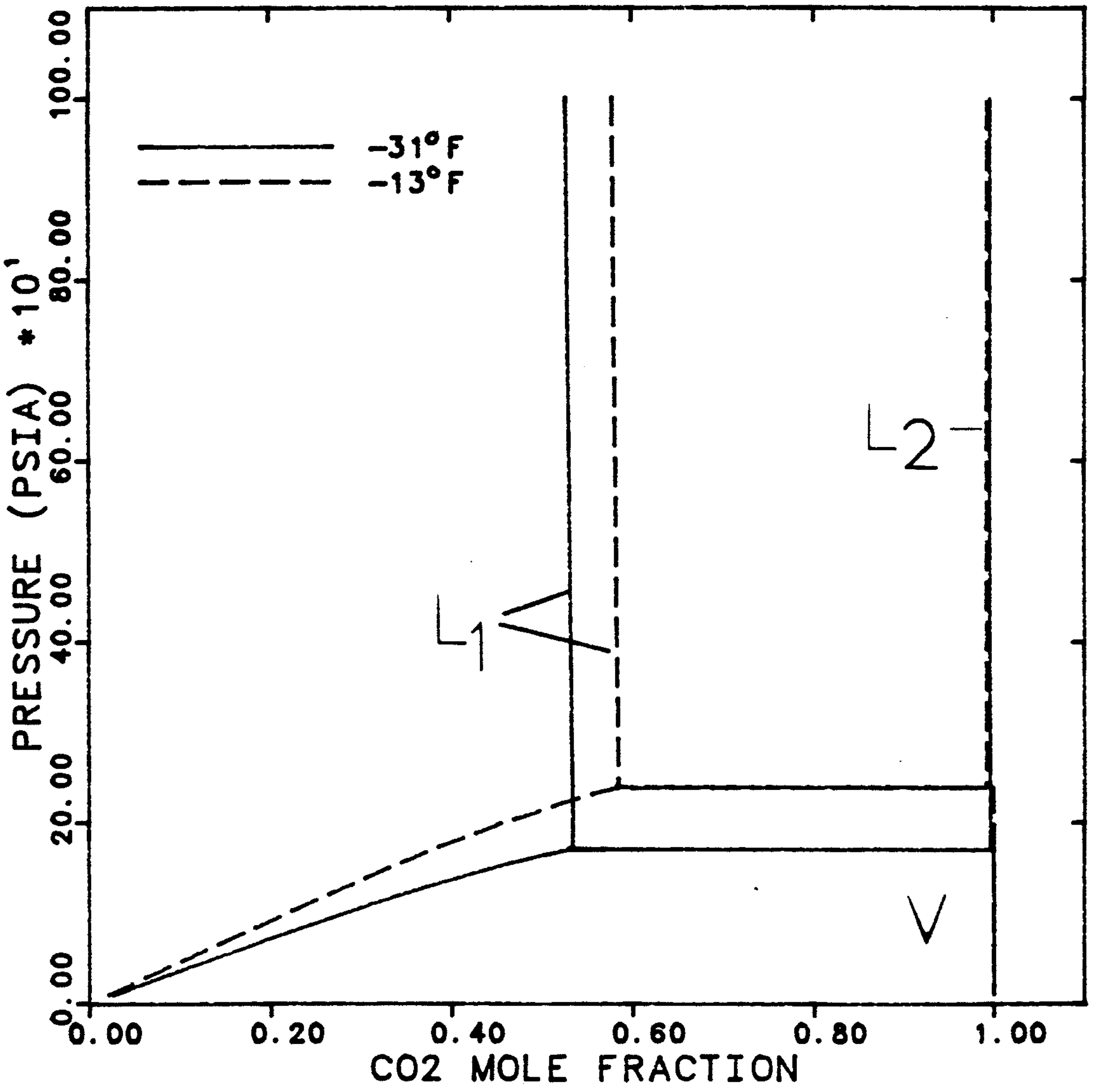


FIGURE 6.5 COMPUTED PHASE EQUILIBRIA FOR THE BINARY MIXTURE OF CO2 AND N-DECANE SHOWING LIQUID-LIQUID AND LIQUID-VAPOR BEHAVIOR.



which was taken from Nagy et al. At low pressures, liquid-vapor equilibrium phases are formed. At the pressure of about 170 psia and 239 psia, three equilibrium phases coexist. At pressures higher than the three-phase pressure, liquid-liquid phase equilibria is found. The composition of the two liquid phases stays virtually the same over wide changes in pressure. Although no comparison is made of calculated equilibrium phase compositions with experimental data, the occurrence of the three-phase pressure is relatively close to that indicated in Figure 2.1 for these temperatures.

In Figure 6.6 is shown the predicted phase behavior for $CO_2$ and normal-decane at 220°F and 100°F using the binary interaction coefficient of 0.115. At 220°F, only liquid-vapor equilibrium appeared which is in agreement with experimental data. At 100°F, however, liquid-vapor and liquid-liquid equilibrium were predicted which is contrary to experimental evidence. The data of Reamer and Sage [R1] show only liquid-vapor equilibrium. Clearly, according to data in Figure 2.1, a three-phase pressure should not occur for temperatures above about -10°F.

In Table 6.3, calculated equilibrium phase compositions for the $CO_2$-n-decane binary are listed. The phase compositions are compared for two different values of the binary interaction coefficient at 220°F and 2300 psia.

FIGURE 6.6 PRESSURE—COMPOSITION DIAGRAM FOR THE CO2 AND N—DECANE BINARY MIXTURE. CALCULATED PHASE ENVELOPE AT HIGHER TEMPERATURES FOR A BINARY INTERACTION COEFFICIENT OF 0.115.

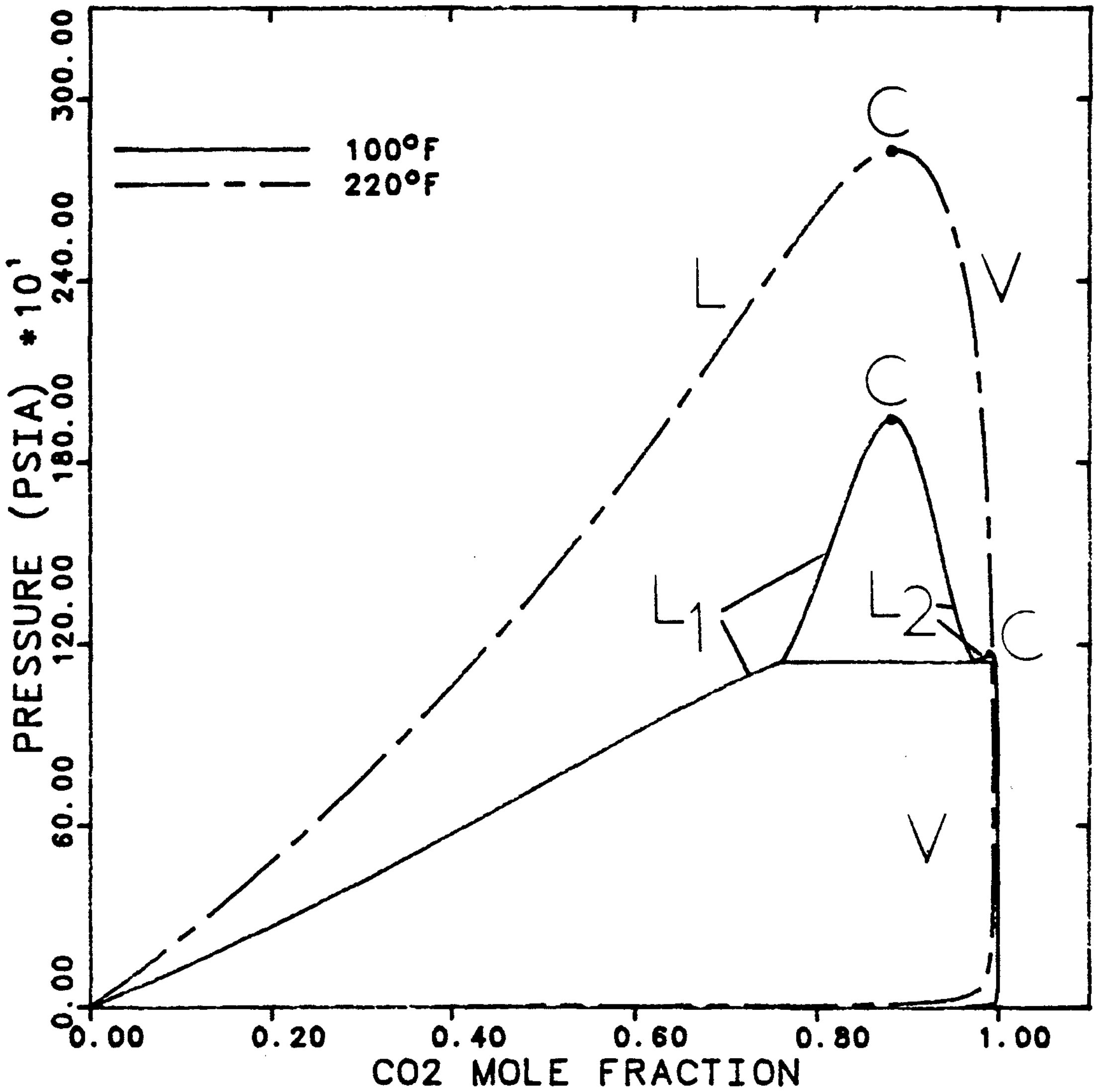


Table 6.3

Calculated equilibrium phase compositions for $CO_2$ and n-decane at 220°F and 2300 psia for two values of the binary interaction coefficient.

A. Binary Interaction Coefficient $\delta_{ij} = 0.115$

| Component | Phase Composition<br>Vapor | <br>Liquid | K-value |
|---|---|---|---|
| $CO_2$ | .97033 | .72197 | 1.3440 |
| n-decane | .02967 | .27803 | 0.10672 |

B. Binary Interaction Coefficient $\delta_{ij} = 0.05$

| Component | Phase Composition<br>Vapor | <br>Liquid | K-value |
|---|---|---|---|
| $CO_2$ | 0.95246 | 0.86828 | 1.0970 |
| n-decane | 0.04754 | 0.13172 | 0.36092 |

The value of 0.115 for the binary interaction coefficient was taken from the literature and the value of 0.05 was chosen sufficiently small to assure that only liquid-vapor behavior was predicted at higher temperatures. Clearly, changing the value of the binary interaction parameter had a noticeable effect on the equilibrium phase compositions, as well as phase densities, which are listed in Table 6.4 for these same conditions. At this temperature and pressure, the mixture is approaching the critical point. For the smaller binary interaction coefficient, the properties of the liquid and vapor phases are closer than for the larger value, and, therefore, the predicted critical point would occur at a lower pressure.

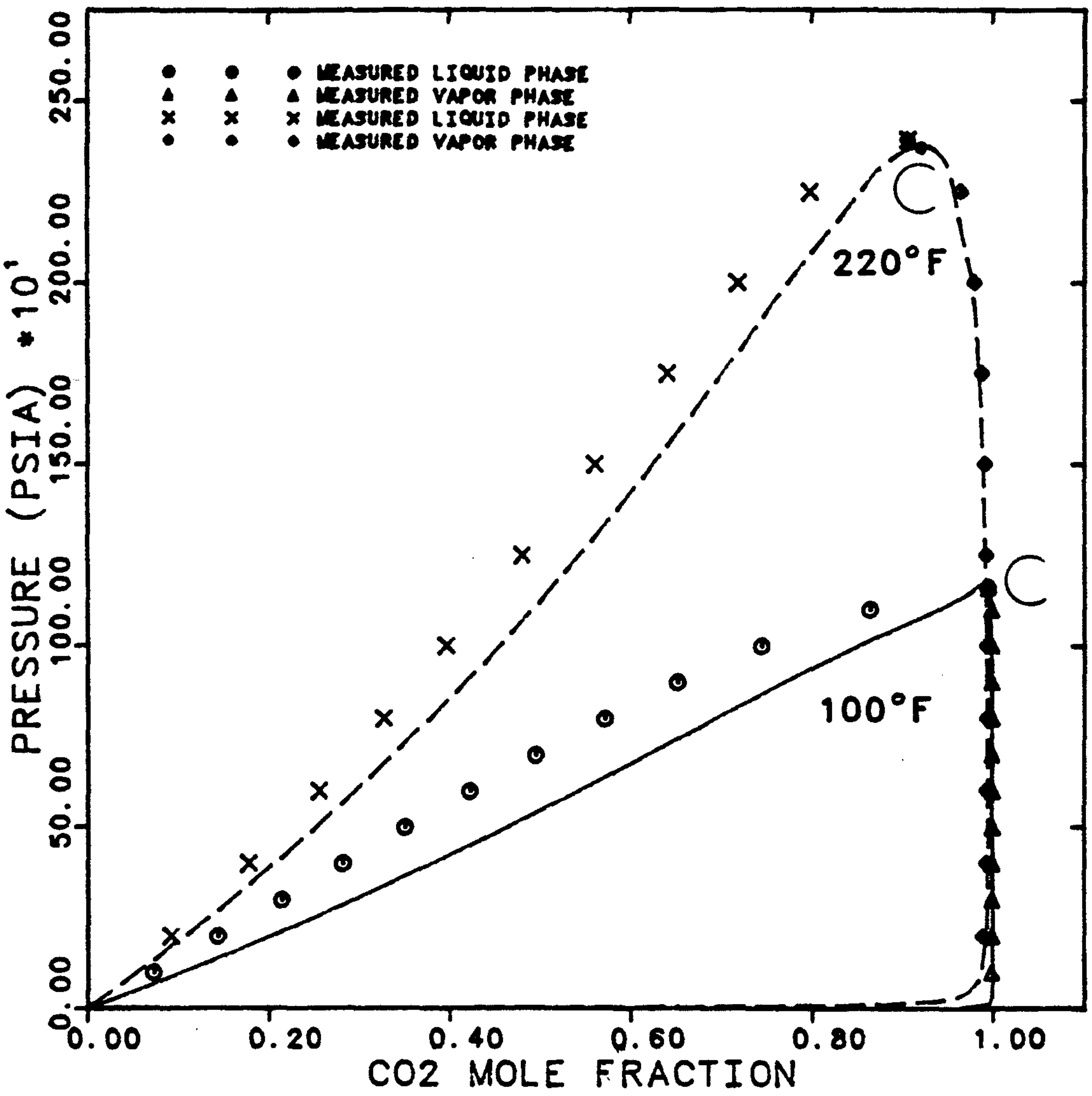
A comparison is made between calculated and experimental equilibrium phase compositions for $CO_2$ and n-decane in the pressure-composition diagram of Figure 6.7 using a binary interaction coefficient of 0.05. The experimental data are from Reamer and Sage. Although the critical point and vapor phase compositions compare quite well with the experimental data, the calculated liquid phase compositions vary a great deal from the experimental values for both temperatures. Note that only liquid-vapor behavior is predicted at 100°F using 0.05 for the binary interaction coefficient.

Table 6.4

Calculated phase densities and compressibility factors for $CO_2$ and n-decane at 220°F and 2300 psia for two values of the binary interaction coefficient.

| | $\delta_{ij} = 0.115$ | | $\delta_{ij} = 0.05$ | |
|---|---|---|---|---|
| | Vapor | Liquid | Vapor | Liquid |
| Compressibility factor | 0.60420 | .56014 | .55584 | .49368 |
| Phase density (lbs/cu ft) | 24.49 | 40.16 | 27.62 | 36.38 |

FIGURE 6.7 PRESSURE VERSUS COMPOSITION DIAGRAM FOR THE CO2 AND N-DECANE MIXTURE. COMPARISON OF CALCULATED AND MEASURED PHASE COMPOSITION AT TWO TEMPERATURES.



Predictions of $CO_2$ and n-decane phase behavior are shown in Figure 6.8 for four values of the binary interaction coefficient at 100°F to examine the effect of changing the value of this parameter. Only liquid-vapor behavior was predicted for a value of 0.095, while for 0.10, liquid-liquid behavior was predicted in addition to liquid-vapor behavior. The value of 0.095 is slightly lower than literature values, but given the range of reported binary interaction parameters, it is not unreasonable. Figure 6.9 shows an expanded view of the area around the three-phase pressures. As the diagram shows, once a liquid-liquid region forms, increasing the value of the binary interaction coefficient produces an increase in the size of the region. Also, the pressure at which the $L_2$-V critical point occurs decreases as the binary interaction coefficient increases.

Several runs were made to examine the effect of perturbations in component critical properties on phase behavior predictions. As Figures 6.10, 6.11, and 6.12 illustrate, relatively small changes in input data produced large changes in the phase behavior.

In Figure 6.10, only the acentric factor for n-decane was changed, from 0.586 to 0.484, using 0.115 for the binary interaction coefficient. As shown, the liquid-liquid region decreases in size. The pressure at which the $L_2$-V critical point occurs increases, as was the case

FIGURE 6.8 PRESSURE–COMPOSITION DIAGRAM FOR THE MIXTURE OF CO2 AND N–DECANE AT 100°F. CALCULATED PHASE ENVELOPE FOR SEVERAL VALUES OF THE BINARY INTERACTION COEFFICIENT.

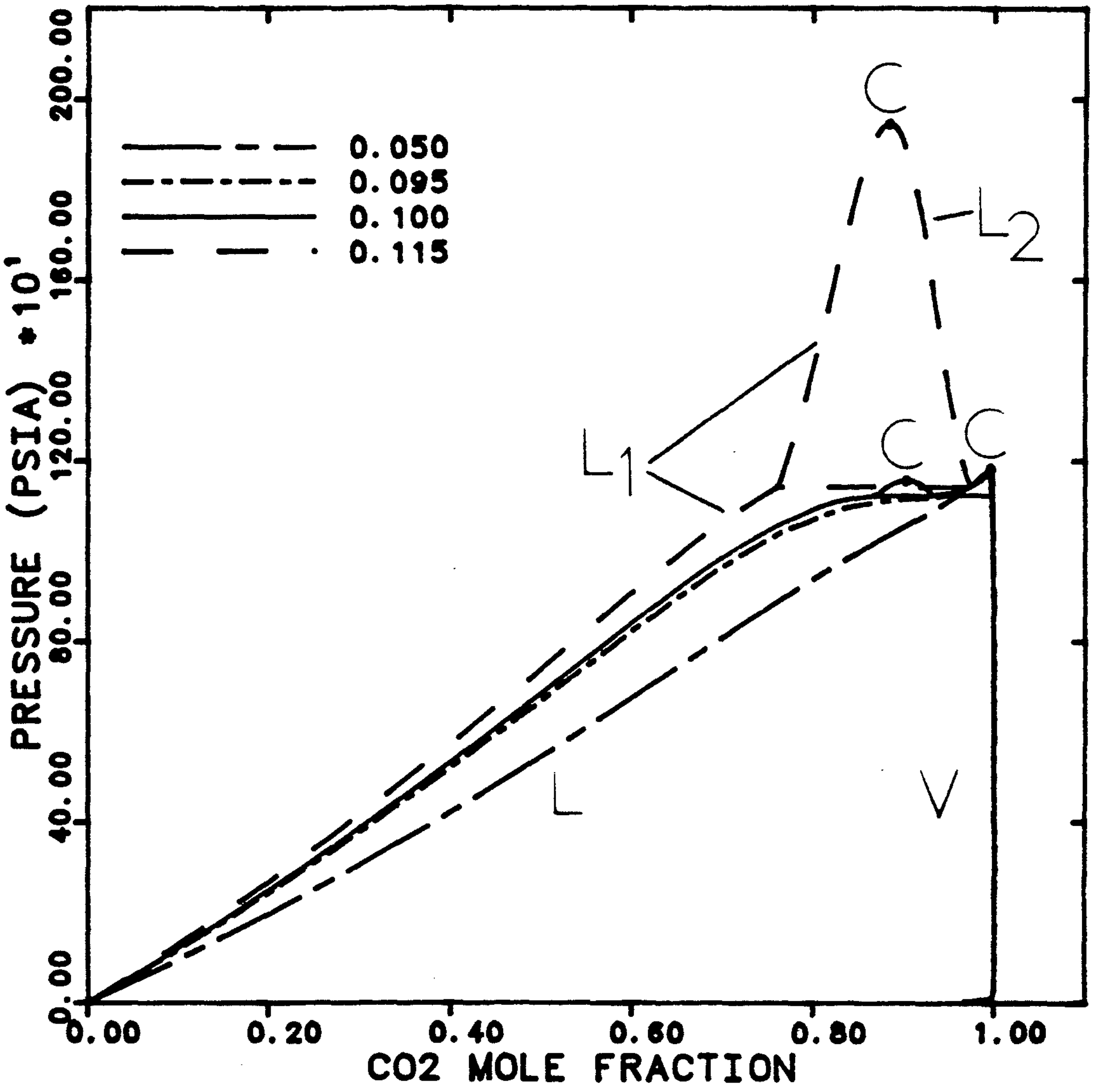


FIGURE 6.9 PRESSURE–COMPOSITION DIAGRAM FOR THE MIXTURE OF CO2 AND N–DECANE AT 100°F. CALCULATED PHASE ENVELOPE FOR SEVERAL VALUES OF THE BINARY INTERACTION COEFFICIENT.

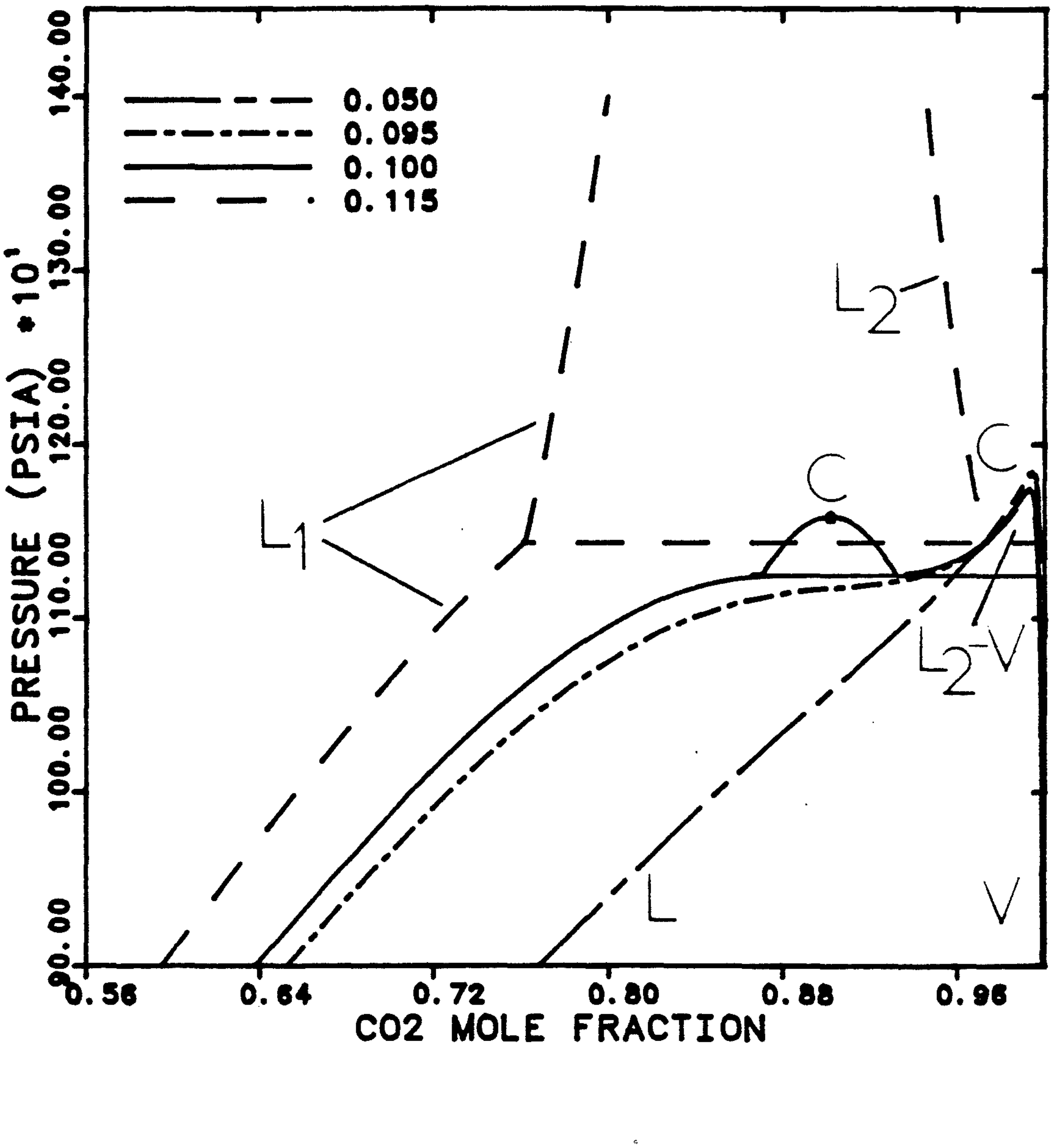


FIGURE 6.10 PRESSURE-COMPOSITION DIAGRAM FOR THE CO2 AND N-DECANE MIXTURE AT 100°F. CALCULATED PHASE ENVELOPE FOR TWO VALUES OF THE N-DECANE ACENTRIC FACTOR.

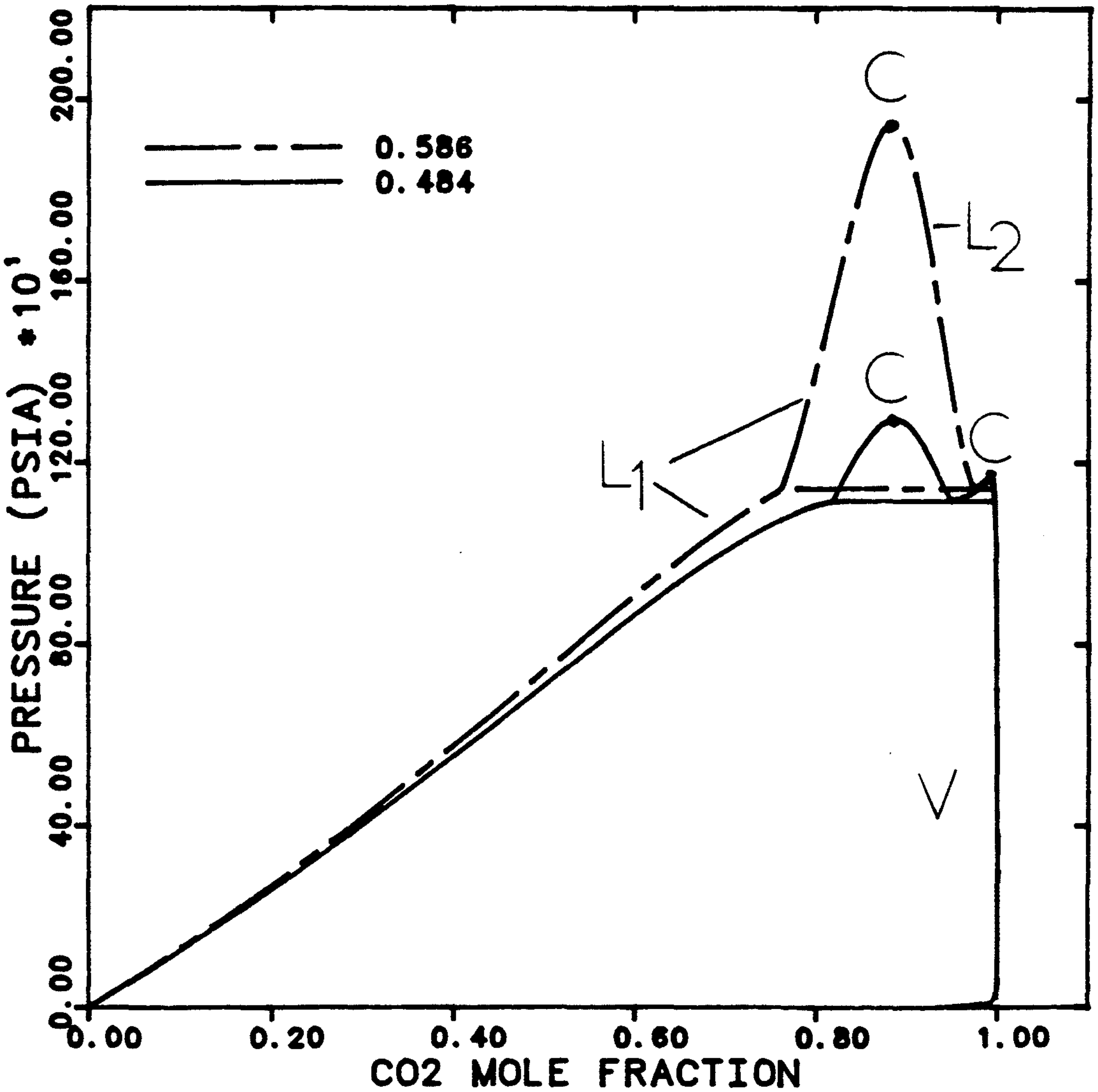


FIGURE 6.11 PRESSURE–COMPOSITION DIAGRAM FOR THE CO2 AND N–DECANE MIXTURE AT 100°F. CALCULATED PHASE ENVELOPE FOR TWO VALUES OF THE N–DECANE CRITICAL TEMPERATURE.

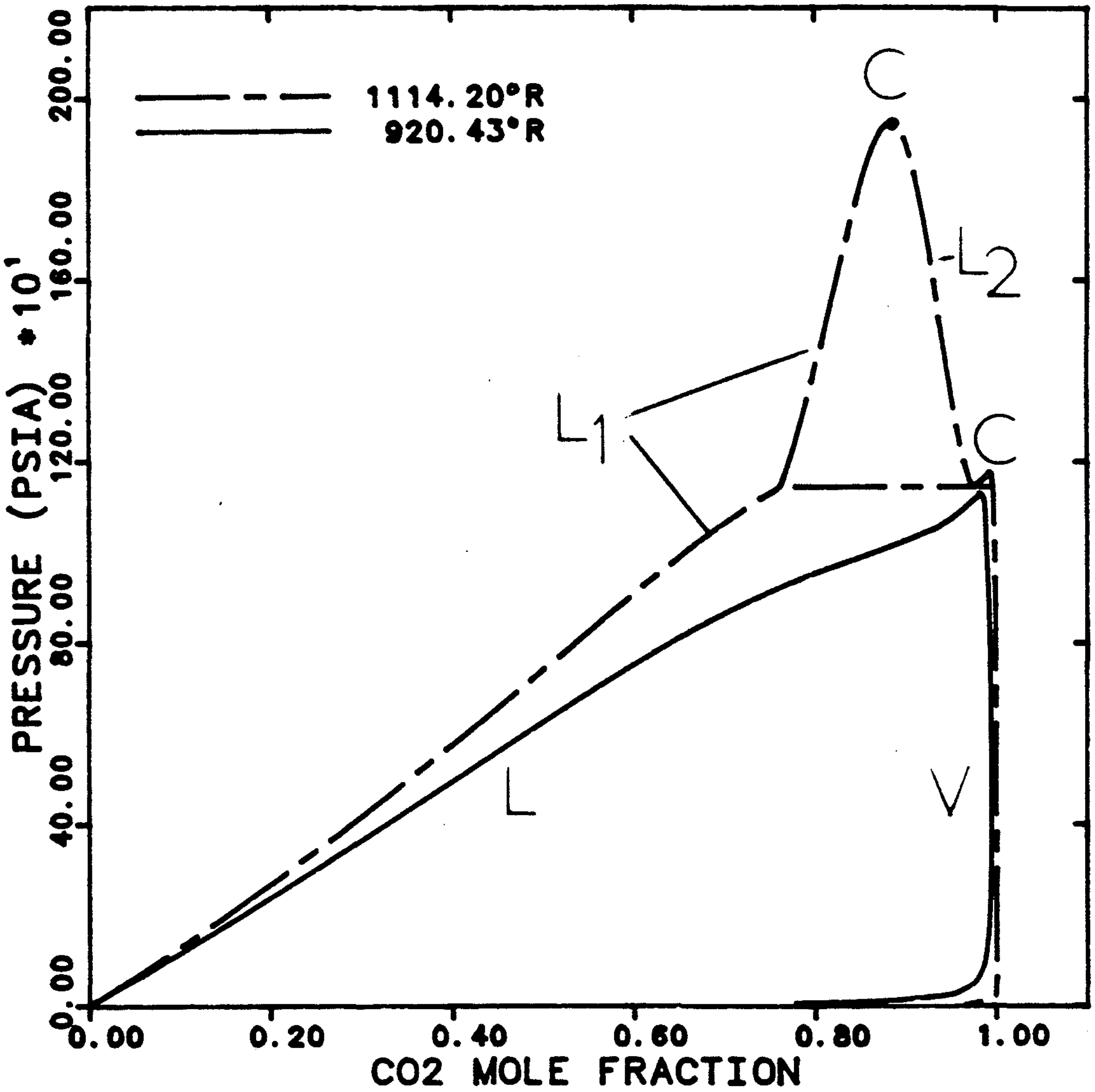
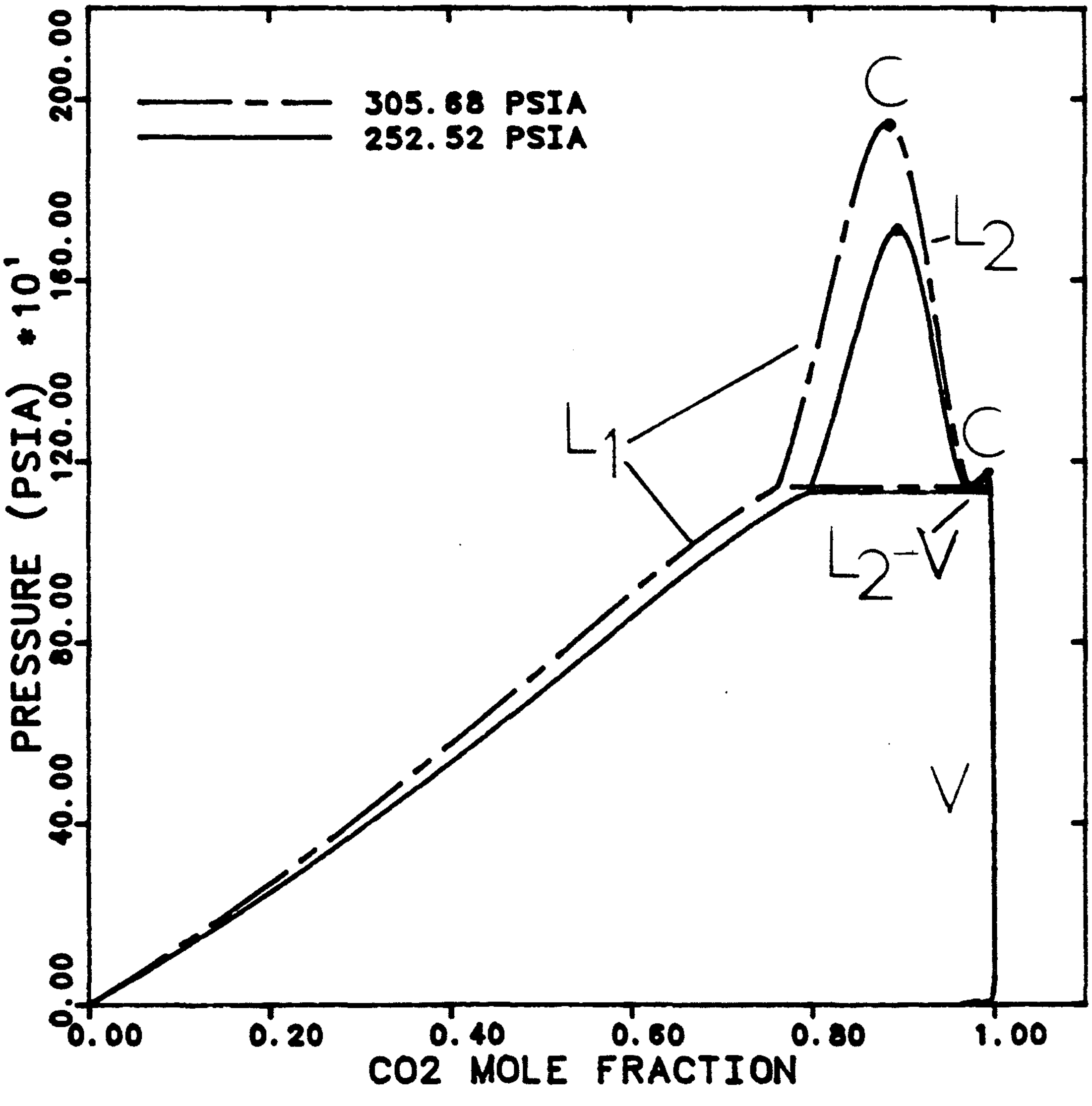


FIGURE 6.12 PRESSURE–COMPOSITION DIAGRAM FOR THE CO2 AND N–DECANE MIXTURE AT 100°F. CALCULATED PHASE ENVELOPE FOR TWO VALUES OF THE N–DECANE CRITICAL PRESSURE.



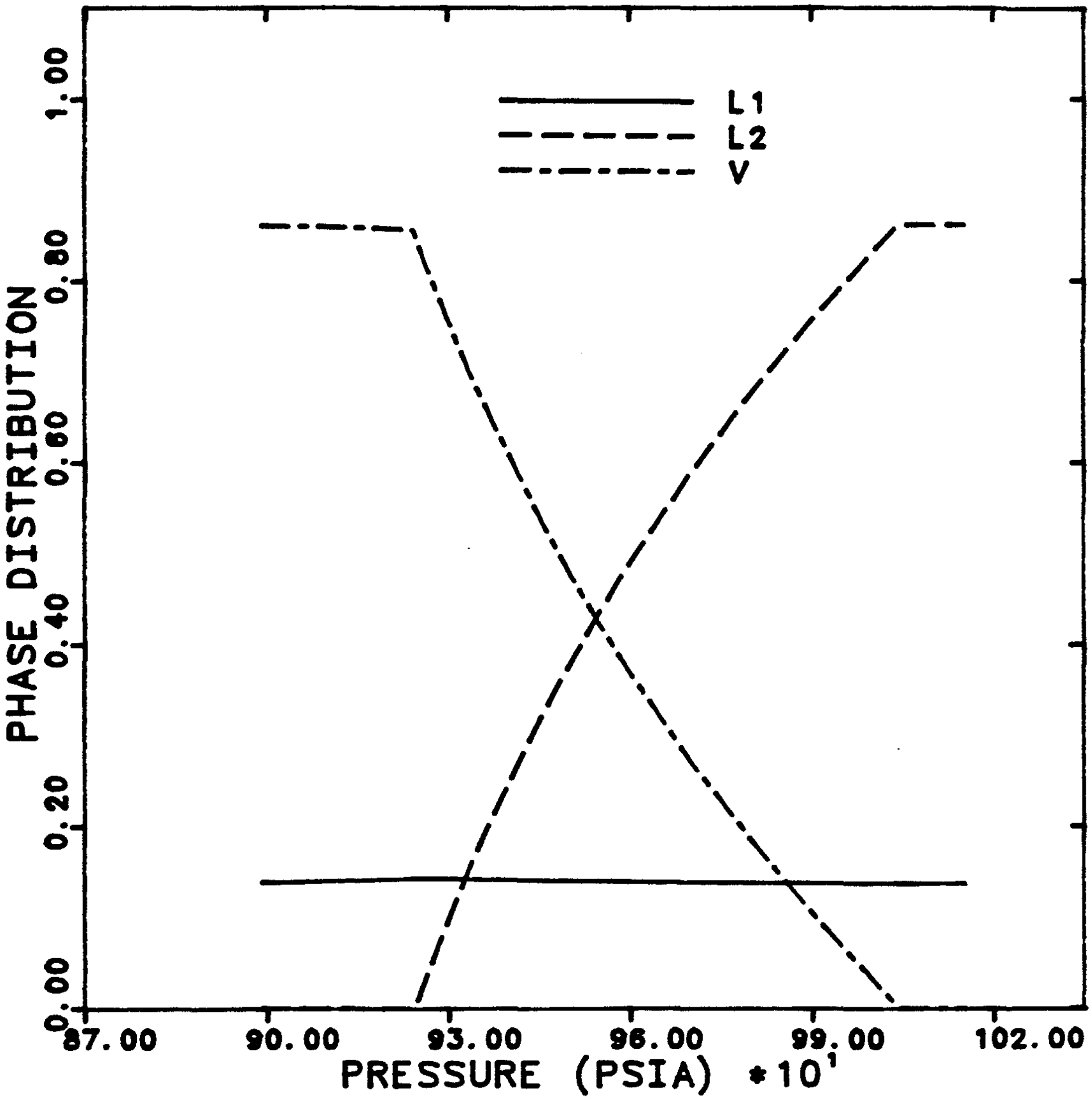
when the binary interaction coefficient decreased in value. Altering only the n-decane critical temperature from 1114.2°R to 920.43°R resulted in the changes shown in Figure 6.11. With the reduced value for the critical temperature, only liquid-vapor equilibria is predicted. Reducing the critical pressure of n-decane from 305.68 psia to 252.52 psia alters the phase behavior somewhat as shown in Figure 6.12. However, perturbing the critical pressure by the same relative amount has a lesser effect than do changes in other input values as shown above.

### 6.3 $CO_2$-$CH_4$-n-$C_{16}$ Ternary Mixture

A ternary mixture of $CO_2$, methane, and n-hexadecane is capable of forming up to three phases at equilibrium. This ternary was examined at 70°F for both two-phase and three-phase behavior. Binary interaction coefficients were taken from Risnes et al.

If for a given temperature the overall composition is fixed and pressure allowed to vary, the relative amounts of each phase, or the phase distribution, change. This is shown in Figure 6.13. The temperature is 70°F and the overall composition is fixed at 0.9 mole percent $CO_2$ with equal amounts of $CH_4$ and n-$C_{16}$. Below about 925 psia, a liquid phase exists in equilibrium with a vapor phase. Between 925 and 1003 psia, three phases coexist--a $CO_2$ lean

FIGURE 6.13 COMPUTED PHASE DISTRIBUTION VERSUS PRESSURE FOR THE MIXTURE OF CO2, CH4, AND N-C16 AT 70°F AND FOR FIXED OVERALL COMPOSITION.



liquid ($L_1$), a $CO_2$-rich liquid ($L_2$), and a vapor phase (V). Phase compositions and other computed phase properties at several pressures in the three-phase region are listed in Tables 6.5 to 6.8. The second liquid, $L_2$, grows while the vapor shrinks as pressure increases. Above 1003 psia, only the two liquids remain. The relative amount of the $CO_2$-lean phase changes only slightly throughout the entire pressure range shown.

In Figure 6.14 is shown a ternary diagram for this mixture at 70°F and 700 psia. At this temperature and pressure, $CO_2$ and $CH_4$ are miscible in all proportions. $CO_2$ and n-$C_{16}$ have a miscibility gap where a liquid and vapor phase are in equilibrium. The $CH_4$ and n-$C_{16}$ binary also has a miscibility gap. For ternary mixtures, a two-phase liquid-vapor region extends across the diagram. The vapor phase contains an extremely small amount of n-$C_{16}$.

As the pressure is increased, a three-phase region forms. Figure 6.15 shows the ternary diagram at 900 psia. Now, the $CO_2$-n-$C_{16}$ binary shows liquid-liquid equilibrium. The $CO_2$-$CH_4$ binary is no longer miscible in all proportions. There is a small miscibility gap at high $CO_2$ concentrations, where a vapor is in equilibrium with liquid. Table 6.9 lists the phase compositions and phase densities for this binary. In addition to the large liquid-vapor region, a small three-phase region forms at high $CO_2$ concentrations.

## Table 6.5

Phase compositions for $Co_2$, $CH_4$, and n-$C_{16}$ mixture at 70°F and 945 psia.

| | Phase Composition (mole fraction) | | |
|---|---|---|---|
| Component | $L_1$ | $L_2$ | V |
| $CO_2$ | 0.6263 | 0.9640 | 0.9336 |
| $CH_4$ | 0.0299 | 0.0313 | 0.0663 |
| n-$C_{16}$ | 0.3438 | 0.0047 | 0.0001 |

## Table 6.6

Phase properties and phase distribution for $CO_2$, $CH_4$, and n-$C_{16}$ mixture at 70°F and 945 psia.

| | $L_1$ | $L_2$ | V |
|---|---|---|---|
| Compressibility factor | 0.41169 | 0.18013 | 0.49637 |
| Phase distribution | 0.14106 | 0.31914 | 0.53980 |
| Average molecular weight | 105.88 | 43.97 | 42.17 |
| density (lbs/cu ft) | 42.79 | 40.61 | 14.13 |

Table 6.7

Phase compostions for $CO_2$, $CH_4$, and n-$C_{16}$ mixture at 70°F and 1000 psia.

| | Phase Composition (mole fraction) | | |
|---|---|---|---|
| Component | $L_1$ | $L_2$ | V |
| $CO_2$ | 0.6191 | 0.9462 | 0.9051 |
| $CH_4$ | 0.0452 | 0.0492 | 0.0948 |
| n-$C_{16}$ | 0.3357 | 0.0046 | 0.0001 |

Table 6.8

Phase properties and phase distribution for $Co_2$, $CH_4$, and n-$C_{16}$ at 70°F and 1000 psia.

| | $L_1$ | $L_2$ | V |
|---|---|---|---|
| Compressibility factor | 0.42886 | 0.19594 | 0.47854 |
| Phase distribution | 0.13751 | 0.83218 | 0.03031 |
| Average molecular weight | 103.99 | 43.47 | 41.37 |
| density (lbs/cu ft) | 42.69 | 39.06 | 15.22 |

FIGURE 6.14 COMPUTED TERNARY DIAGRAM FOR THE MIXTURE OF CO2, CH4, AND N-C16 AT 70°F AND 700 PSIA.

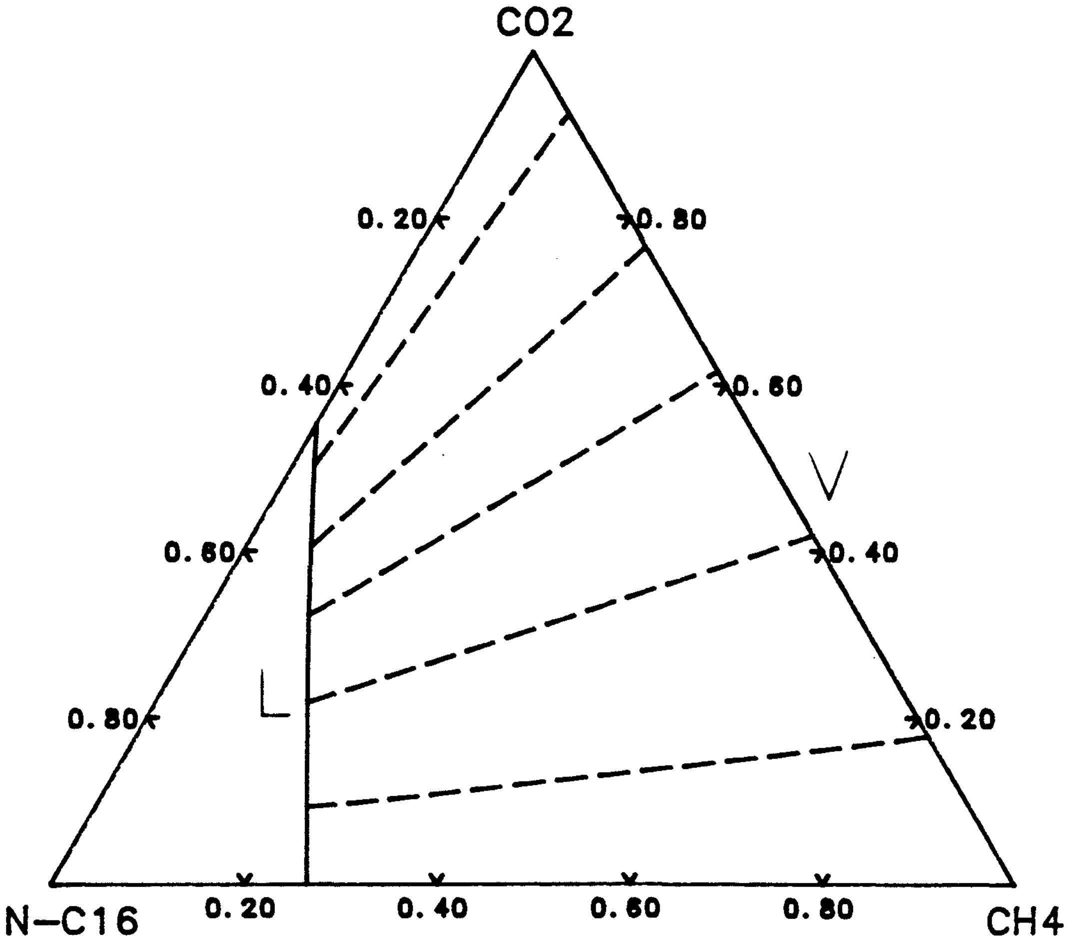


FIGURE 6.15 COMPUTED TERNARY DIAGRAM FOR THE MIXTURE OF CO2, CH4, AND N-C16 AT 70°F AND 900 PSIA.

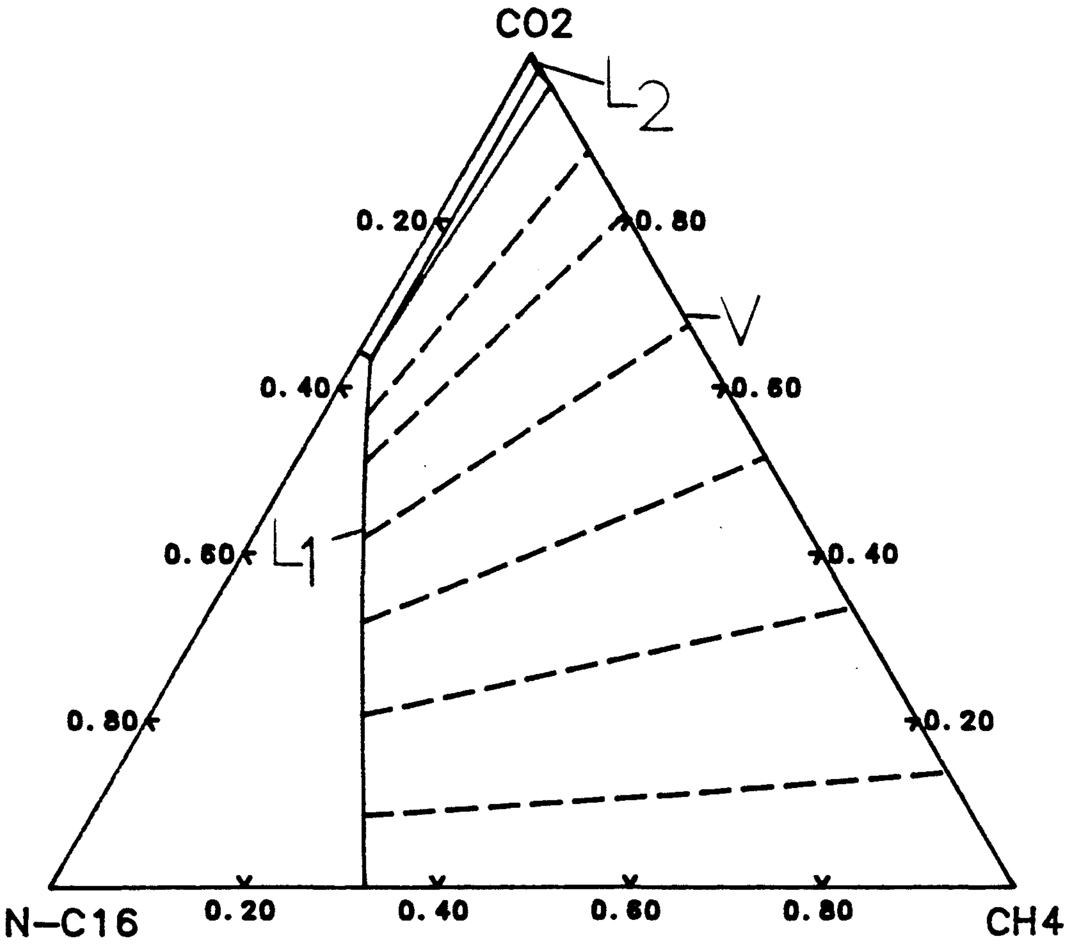


Table 6.9

Phase compositions and densities for the $CO_2$-$CH_4$ binary mixture at 70°F and 900 psia.

| | Composition | |
|---|---|---|
| | Liquid | Vapor |
| $CO_2$ | 0.98585 | 0.96760 |
| $CH_4$ | 0.01415 | 0.03240 |
| density (lbs/cu ft) | 41.76 | 13.67 |

Between the three-phase region and the $CO_2$-n-$C_{16}$ side of the triangle there is a small liquid-liquid region. Notice that the composition of the $CO_2$-rich liquid is very close to the composition of the vapor phase in the three-phase region.

Figures 6.16 and 6.17 show the phase envelopes at 1000 and 1100 psia, respectively. As pressure increases, the three-phase region moves toward higher methane concentrations. As this region moves, the liquid-vapor region decreases in size while the liquid-liquid region increases.

Figure 6.18 shows the ternary diagram at 2000 psia. At this pressure, there is no three-phase region. $CO_2$ and $CH_4$ are again miscible in all proportions. The two-phase region extends as a band across the entire diagram. Tables 6.10 and 6.11 list phase compositions and phase densities for equilibrium phases at high $CO_2$ concentrations and also at low $CO_2$ concentrations. From Table 6.10 it appears that two liquids are in equilibrium. In fact, the $CO_2$-lean liquid phase is less dense than is the $CO_2$-rich liquid phase. However, at low $CO_2$ concentrations, from Table 6.11, it appears as though a liquid is in equilibrium with a vapor. The $CO_2$-rich liquid seemingly transforms into a vapor phase as the $CO_2$ concentration decreases without going through a critical point or three-phase region.

FIGURE 6.16 COMPUTED TERNARY DIAGRAM FOR THE MIXTURE OF CO2, CH4, AND N-C16 AT 70°F AND 1000 PSIA.

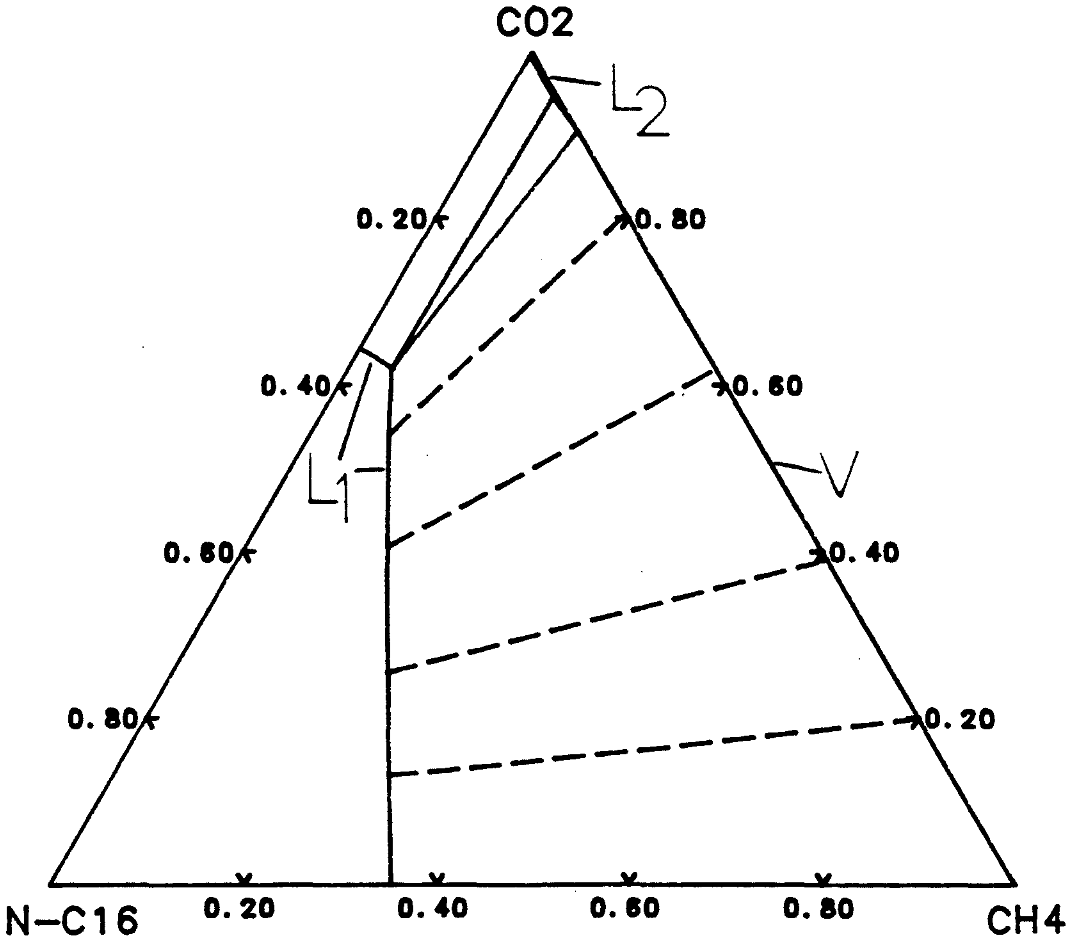


FIGURE 6.17 COMPUTED TERNARY DIAGRAM FOR THE MIXTURE OF CO2, CH4, AND N-C16 AT 70°F AND 1100 PSIA.

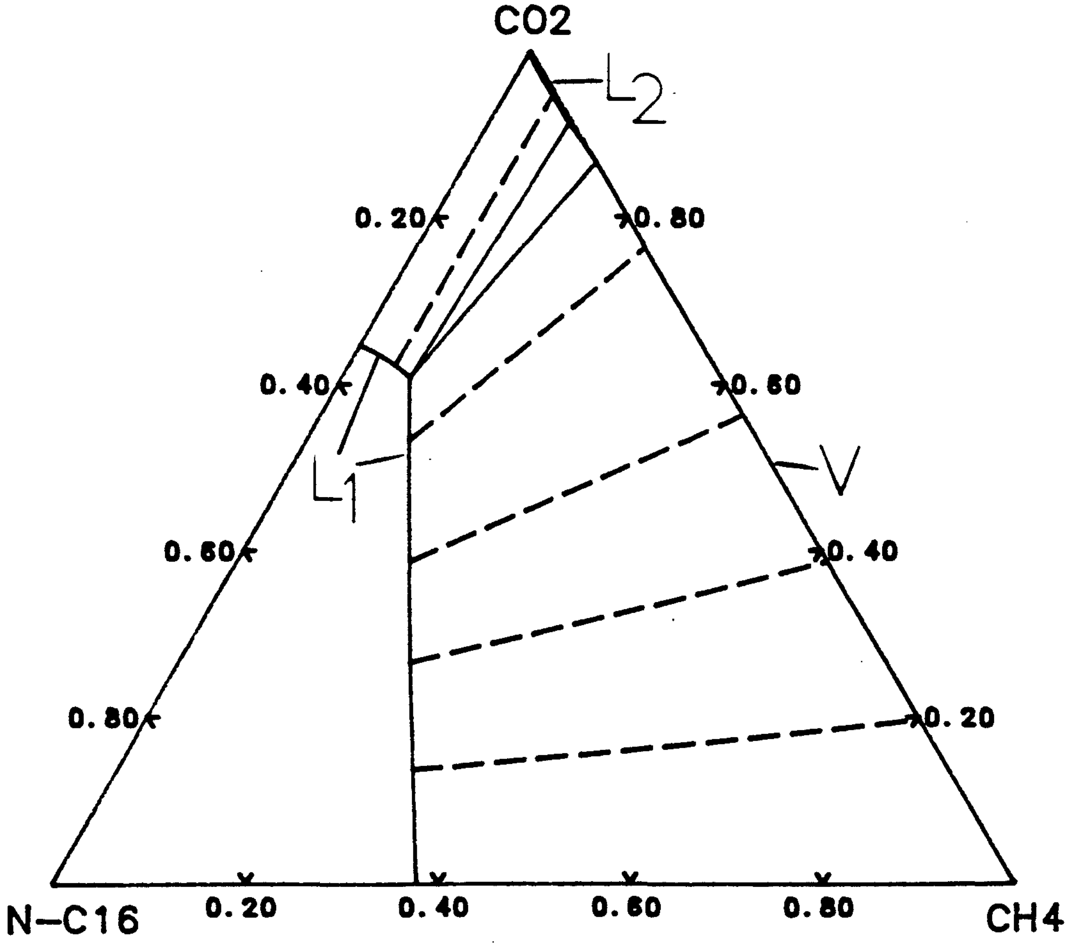


FIGURE 6.18 COMPUTED TERNARY DIAGRAM FOR THE MIXTURE OF CO2, CH4 AND N-C16 AT 70°F AND 2000 PSIA.

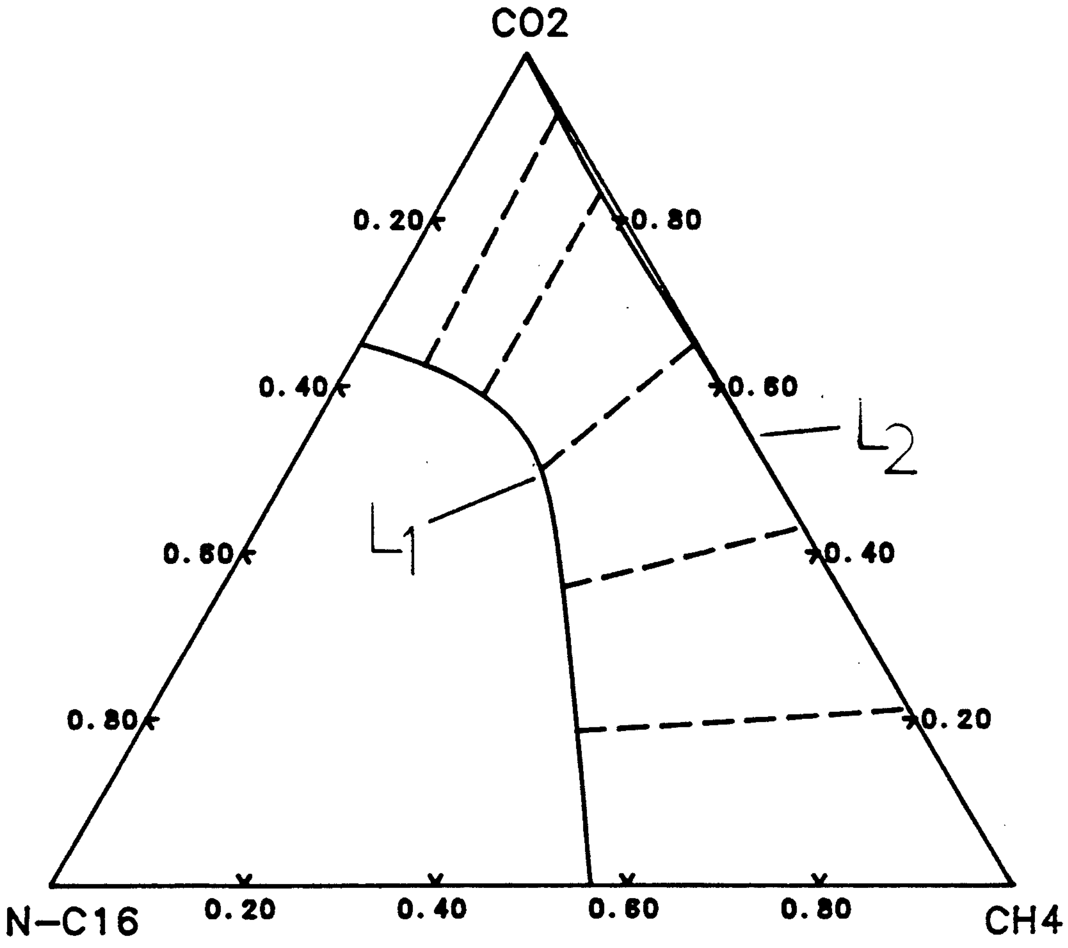


Table 6.10

Phase compositions and densities for the $CO_2$, $CH_4$, and n-$C_{16}$ ternary at 70°F and 2000 psia.

| Component | Composition<br>$L_1$ | <br>$L_2$ |
|---|---|---|
| $CO_2$ | 0.61322 | 0.89378 |
| $CH_4$ | 0.10895 | 0.09903 |
| n-$C_{16}$ | 0.27783 | 0.00719 |
| Compressibility factor | .75149 | .32604 |
| Phase density (lbs/cu ft) | 42.93 | 45.95 |

Table 6.11

Phase compositions and densities for the $CO_2$, $CH_4$, and n-$C_{16}$ ternary at 70°F and 2000 psia.

| Component | Composition<br>$L_1$ | <br>$L_2$ |
|---|---|---|
| $CO_2$ | 0.09384 | 0.10510 |
| $CH_4$ | 0.50889 | 0.89475 |
| n-$C_{16}$ | 0.29727 | 0.00015 |
| Compressibility factor | 0.98852 | 0.77843 |
| Average molecular weight | 105.13 | 17.53 |
| Phase density (lbs/cu ft) | 37.44 | 7.93 |

Figure 6.19 illustrates the movement of the three-phase region as pressure varies. The composition of the $CO_2$-lean liquid stays relatively constant, changing mostly in an increase in the amount of methane contained in the phase. Note that the $CO_2$-rich liquid always has a higher $CO_2$ concentration than does the vapor in the three-phase region.

The effect of pressure on the location of the three-phase region can be illustrated on a pressure-composition diagram. Figure 6.20 shows part of one pressure-composition diagram for pressures at which three phases occur. In this figure, the overall mole fraction of $CO_2$ in the ternary mixture is varied. The ratio of methane to $n\text{-}C_{16}$ is fixed at 30:70. Thus, varying the amount of $CO_2$ in the mixture at a fixed pressure is equivalent to moving along a dilution line that goes from the 30 percent $CH_4$: 70 percent $n\text{-}C_{16}$ point on the $CH_4$-$n\text{-}C_{16}$ side of the triangle to the $CO_2$ apex. Points where the dilution line crosses phase boundaries at a given pressure are cross-plotted on the P-X diagram. For the $CO_2$-$CH_4$-$n\text{-}C_{16}$ mixture, as pressure increases, the dilution line crosses the three-phase boundaries at points which are continually higher in methane concentration. As the amount of methane increases, the concentration of $CO_2$ decreases since the ratio of $CH_4$ to $n\text{-}C_{16}$ has been fixed. The movement of the

FIGURE 6.19 MOVEMENT OF THE COMPUTED THREE-PHASE REGION AS THE PRESSURE VARIES FOR THE MIXTURE OF CO2, CH4, AND N-C16 AT 70°F.

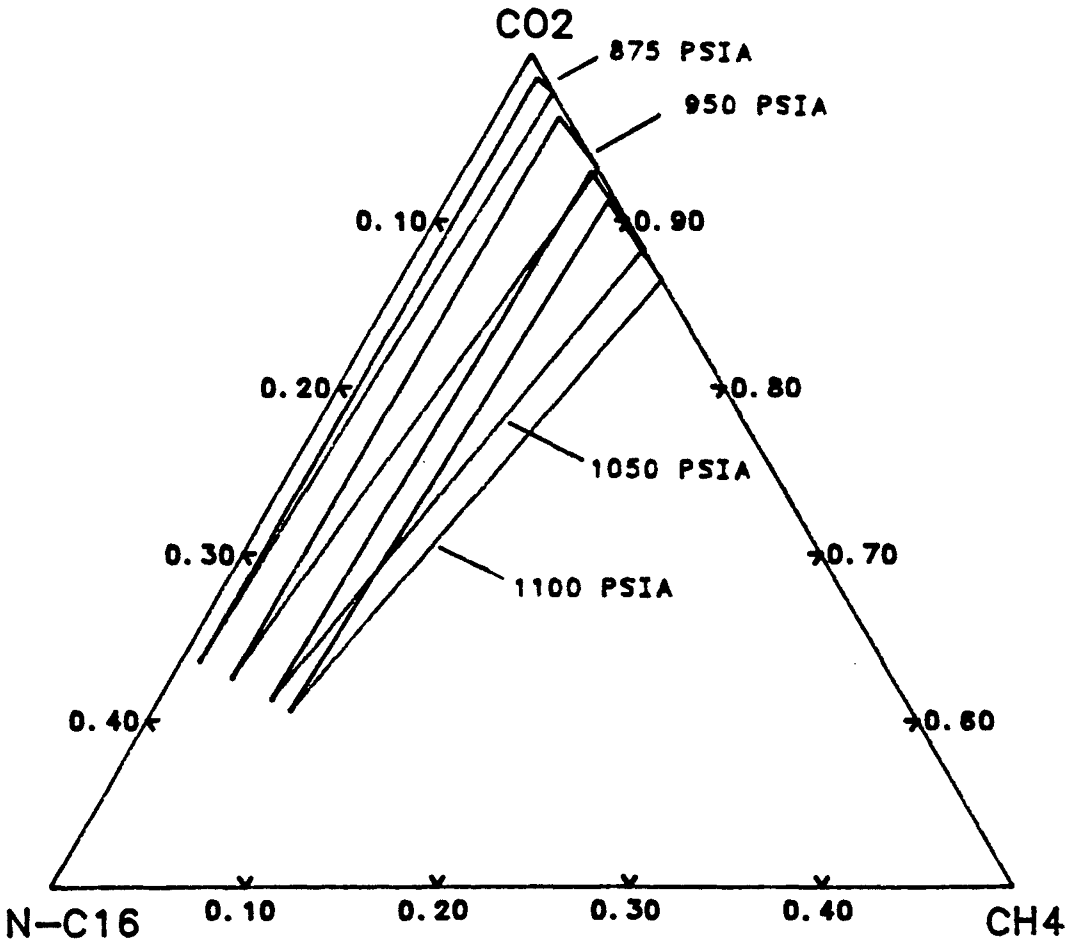
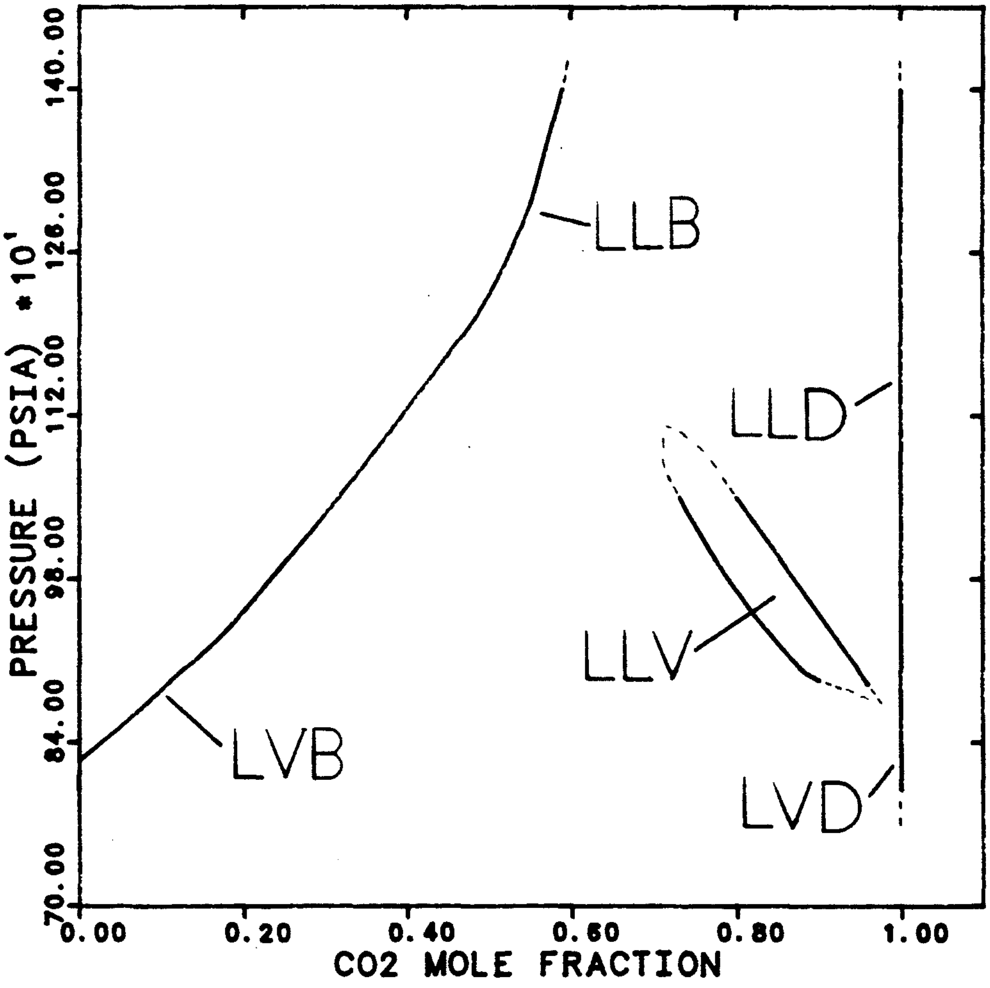


FIGURE 6.20 PRESSURE-COMPOSITION DIAGRAM FOR A MIXTURE OF CO2 WITH CH4 AND N-C16 AT 70°F. THE RATIO OF CH4 TO N-C16 IS FIXED AT 30 : 70.



three-phase region with changes in pressure is therefore reflected in the P-X diagram.

### 6.4 $CO_2$-$C_3H_8$-n-$C_{16}$ Ternary Mixture

The ternary mixture of $CO_2$, propane, and n-$C_{16}$ may from up to three coexisting equilibrium phases. However, the behavior of this mixture differs a great deal from that of the $CO_2$-$CH_4$-n-$C_{16}$ mixture.

Figure 6.21 is a ternary diagram showing a comparison between experimental and calculated values of the compositions in the three-phase region at several different pressures. The calculations were made using the binary interaction coefficients of Risnes et al. for the $CO_2$-n-$C_{16}$ binary, and those of Lawal for the $C_3H_8$-n-$C_{16}$ and $CO_2$-$C_3H_8$ binaries. With these values, the computed phase compositions are far from the experimental compositions. No attempt was made to adjust input values to match the experimental data. Nevertheless, a good qualitative representation was provided.

Figure 6.22 shows the computed ternary diagram for the $CO_2$-$C_3H_8$-n-$C_{16}$ mixture at 70°F and 500 psia. $C_3H_8$ and $CO_2$ show a miscibility gap as does the binary $CO_2$ and n-$C_{16}$. $C_3H_8$ and n-$C_{16}$ are miscible in all proportions. For ternary mixtures, a two-phase vapor-liquid region forms a band across the diagram connecting the $CO_2$-n-$C_{16}$ side with the

FIGURE 8.21 COMPARISON OF THE EXPERIMENTAL AND COMPUTED LOCATION OF THE THREE-PHASE REGION FOR THE MIXTURE OF CO2, C3H8, AND N-C16 AT 70°F.

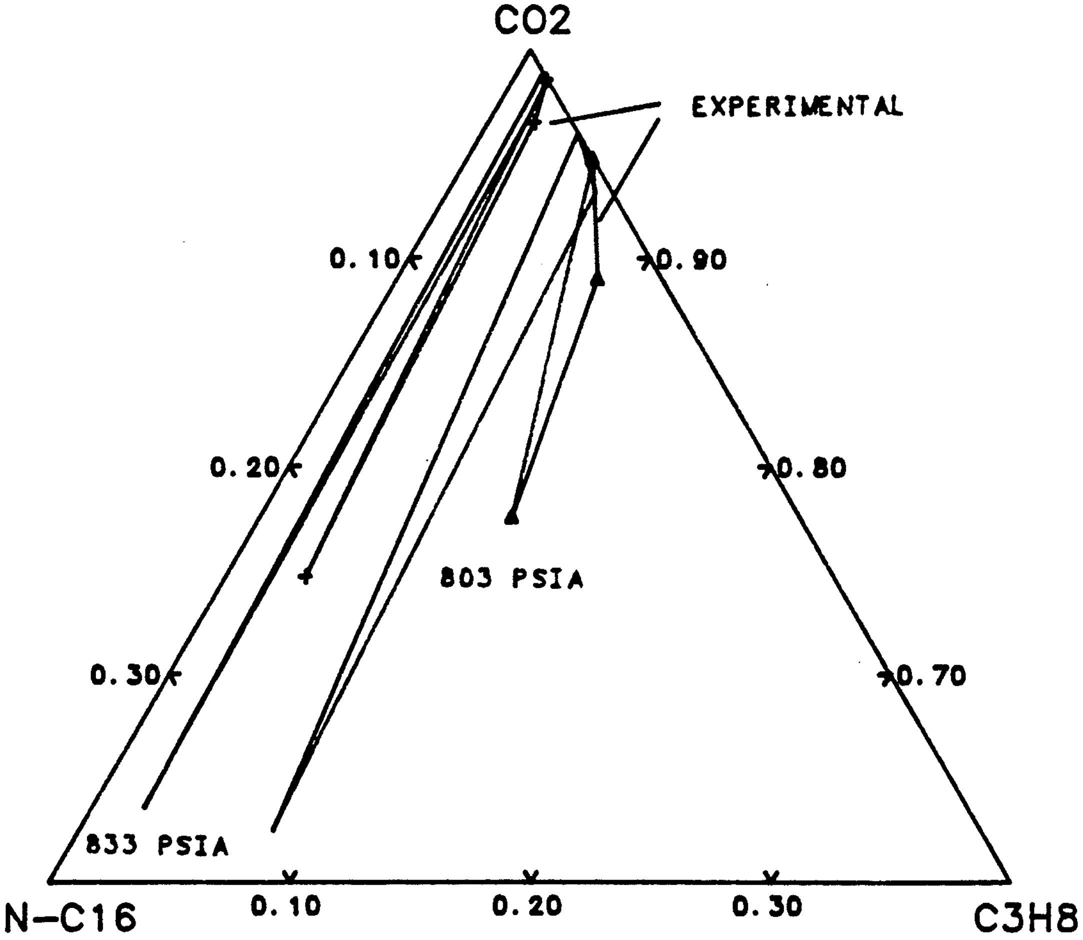
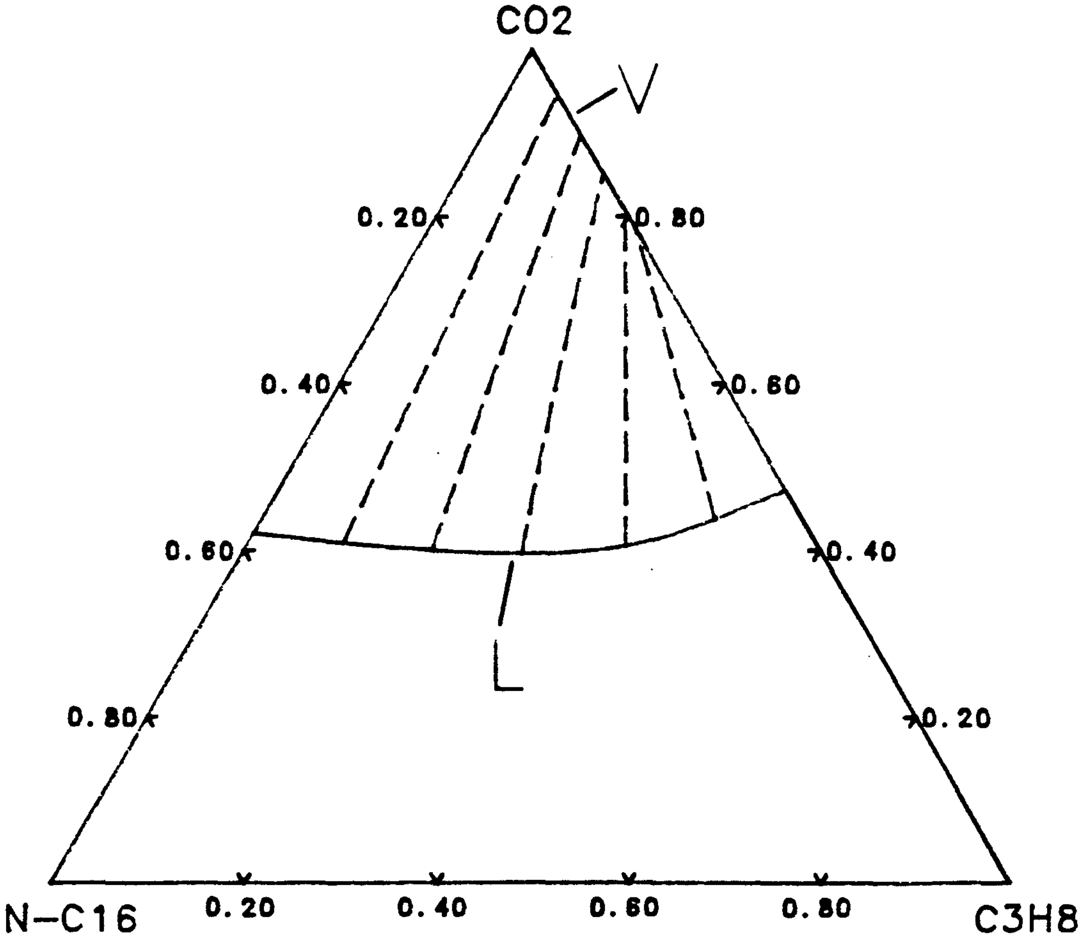


FIGURE 6.22 COMPUTED TERNARY DIAGRAM FOR THE MIXTURE OF CO2, C3H8, AND N-C16 AT 70°F AND 500 PSIA.



$CO_2$-$C_3H_8$ side. As in the $CO_2$-$CH_4$-n-$C_{16}$ mixture, the vapor phase contains only trace amounts of n-$C_{16}$.

Figure 6.23 shows the ternary diagram at a higher pressure of 700 psia. At this temperature and pressure, a three-phase region appears. The $CO_2$-$C_3H_8$ binary has a liquid-vapor miscibility gap as does the $CO_2$-n-$C_{16}$ binary. Phase compositions and densities for the three-phase invariant points are given in Table 6.12. Here, the $CO_2$-rich liquid has a $CO_2$ concentration lower than that of the vapor phase. This is opposite to the behavior of the $CO_2$-$CH_4$-n-$C_{16}$ ternary mixture. A liquid-liquid region also exists, with the liquid phase binodals ending at a critical point.

Figure 6.24 shows the ternary diagram at 795 psia. The appearance of the diagram is similar to that at 700 psia. The liquid-vapor regions and the three-phase region have decreased in size while the liquid-liquid region has gotten larger.

Figure 6.25 shows the ternary diagram at 1000 psia. At this pressure and temperature, the three-phase region has disappeared and all that remains is a liquid-liquid region. $CO_2$ and $C_3H_8$ are miscible in all proportions at these conditions.

Figure 6.26 shows the movement of the three-phase region as pressure varies. As pressure increases, the compositions of the phases increase in $CO_2$ concentration,

FIGURE 6.23 COMPUTED TERNARY DIAGRAM FOR THE MIXTURE OF CO2, C3H8, AND N-C16 AT 70°F AND 700 PSIA.

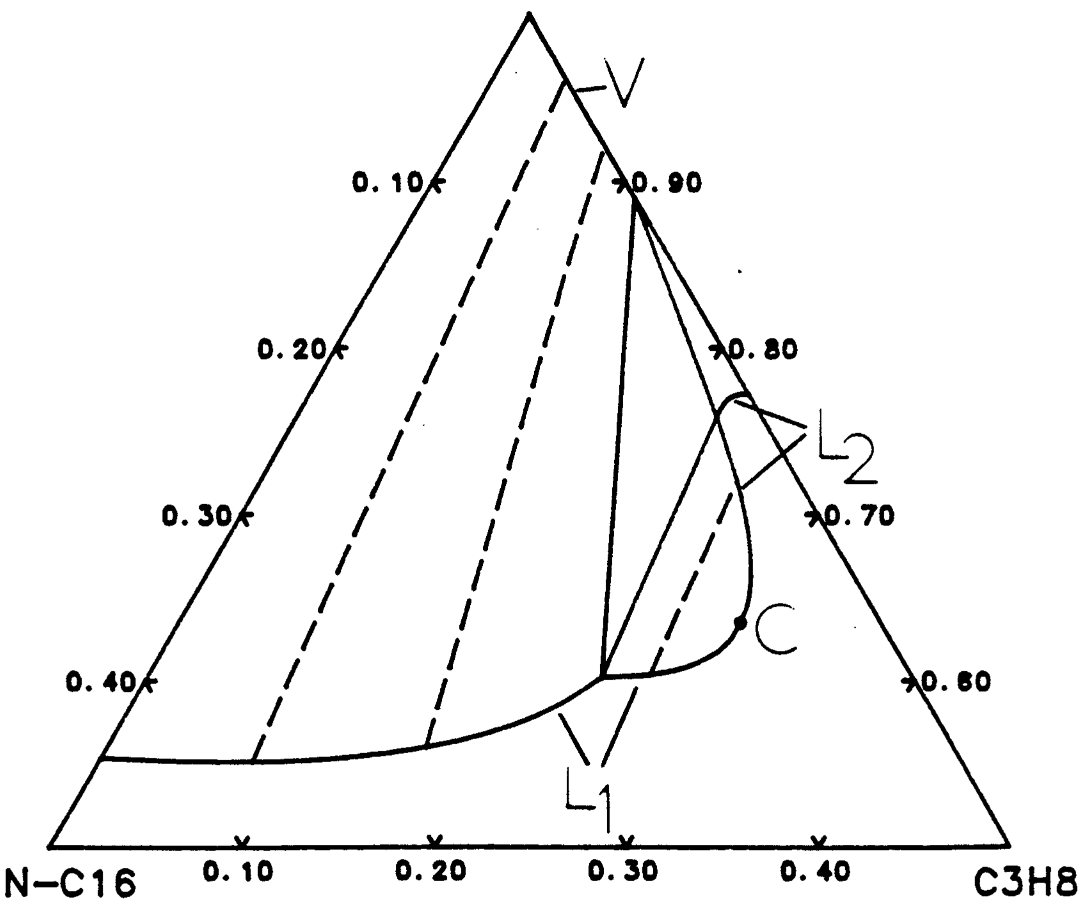


Table 6.12

Phase compositions and densities for the mixture of $CO_2$, $C_3H_8$, and n-$C_{16}$ at 70°F and 700 psia.

| | Composition | | |
|---|---|---|---|
| Component | $L_1$ | $L_2$ | V |
| $CO_2$ | 0.60158 | 0.75827 | 0.89048 |
| $C_3H_8$ | 0.23643 | 0.21802 | 0.10951 |
| n-$C_{16}$ | 0.16199 | 0.02371 | 0.00001 |
| Compressibility factor | 0.21619 | 0.14823 | 0.58678 |
| Average molecular weight | 73.57 | 48.35 | 44.02 |
| Density (lbs/cu ft) | 41.94 | 40.20 | 9.25 |

FIGURE 6.24 COMPUTED TERNARY DIAGRAM FOR THE MIXTURE OF CO2, C3H8, AND N-C16 AT 70°F AND 795.0 PSIA.

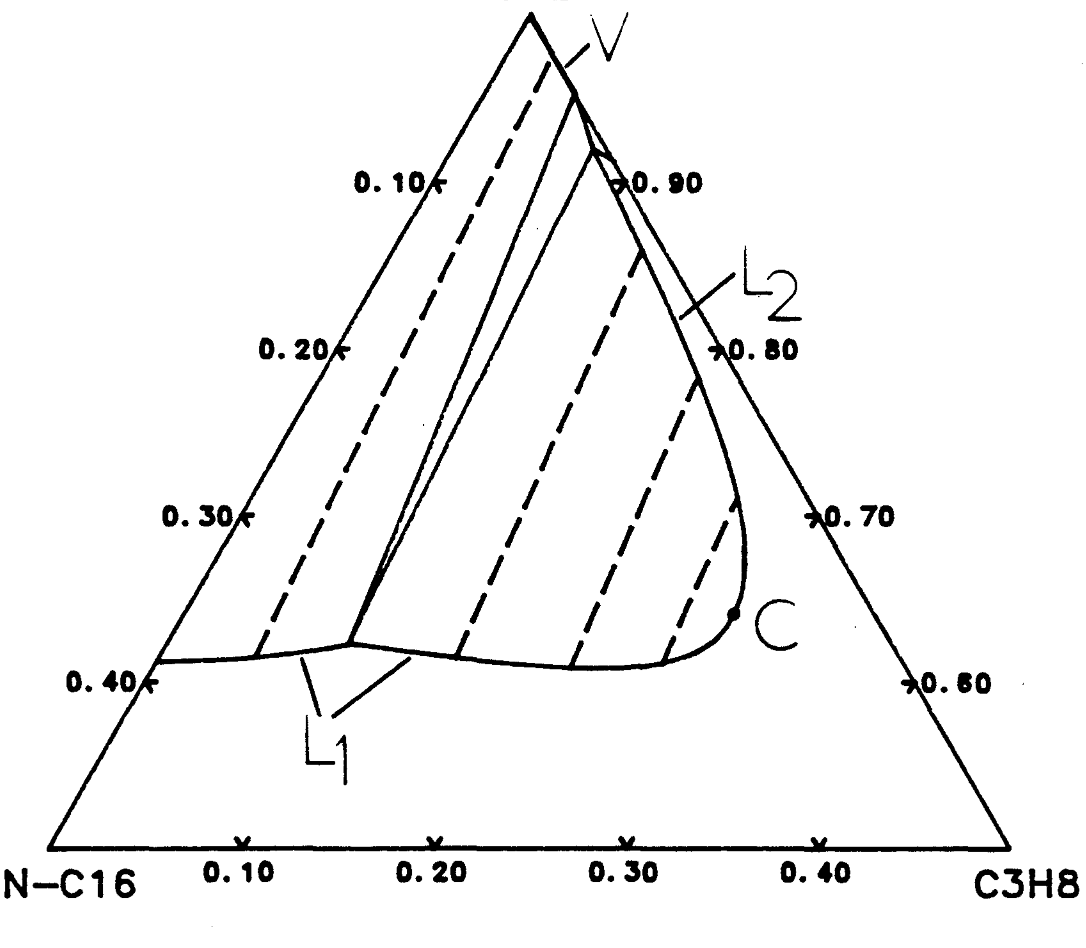


FIGURE 6.25 COMPUTED TERNARY DIAGRAM FOR THE MIXTURE OF CO2, C3H8, AND N-C16 AT 70°F AND 1000.0 PSIA.

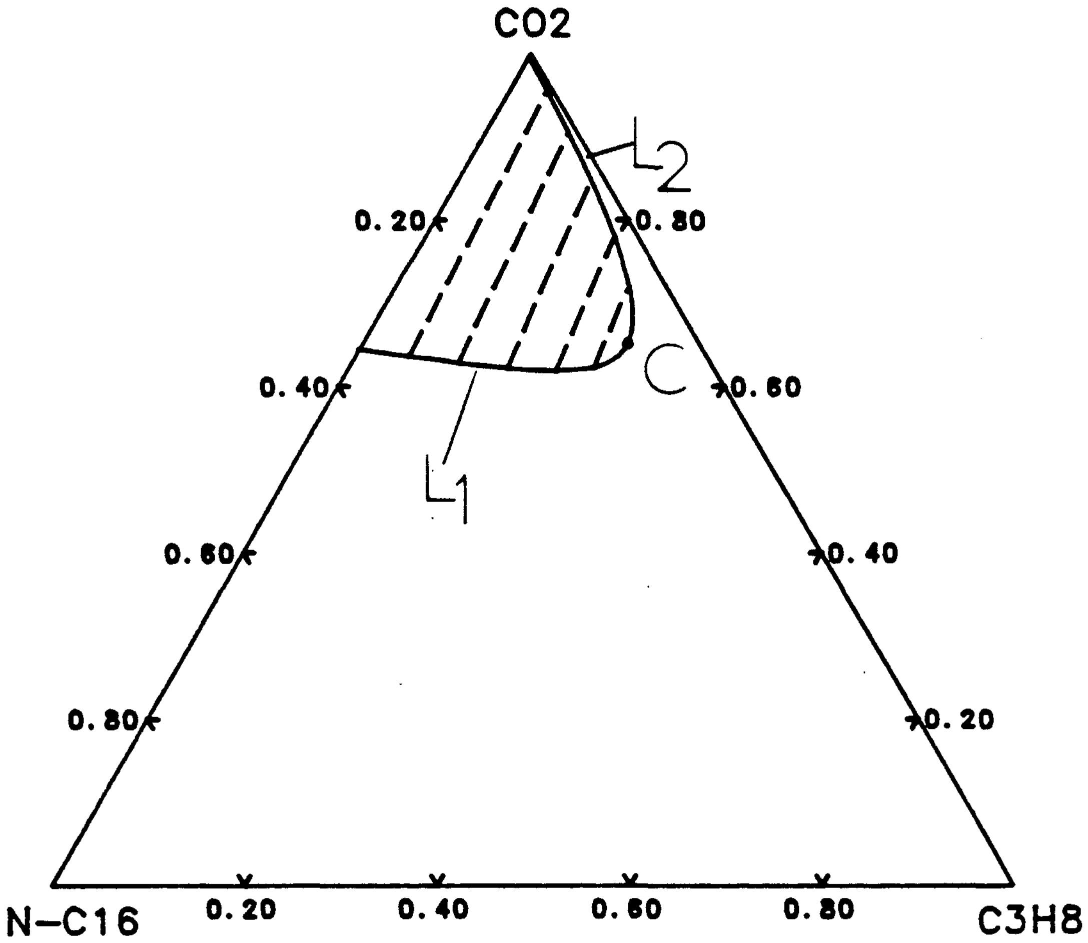
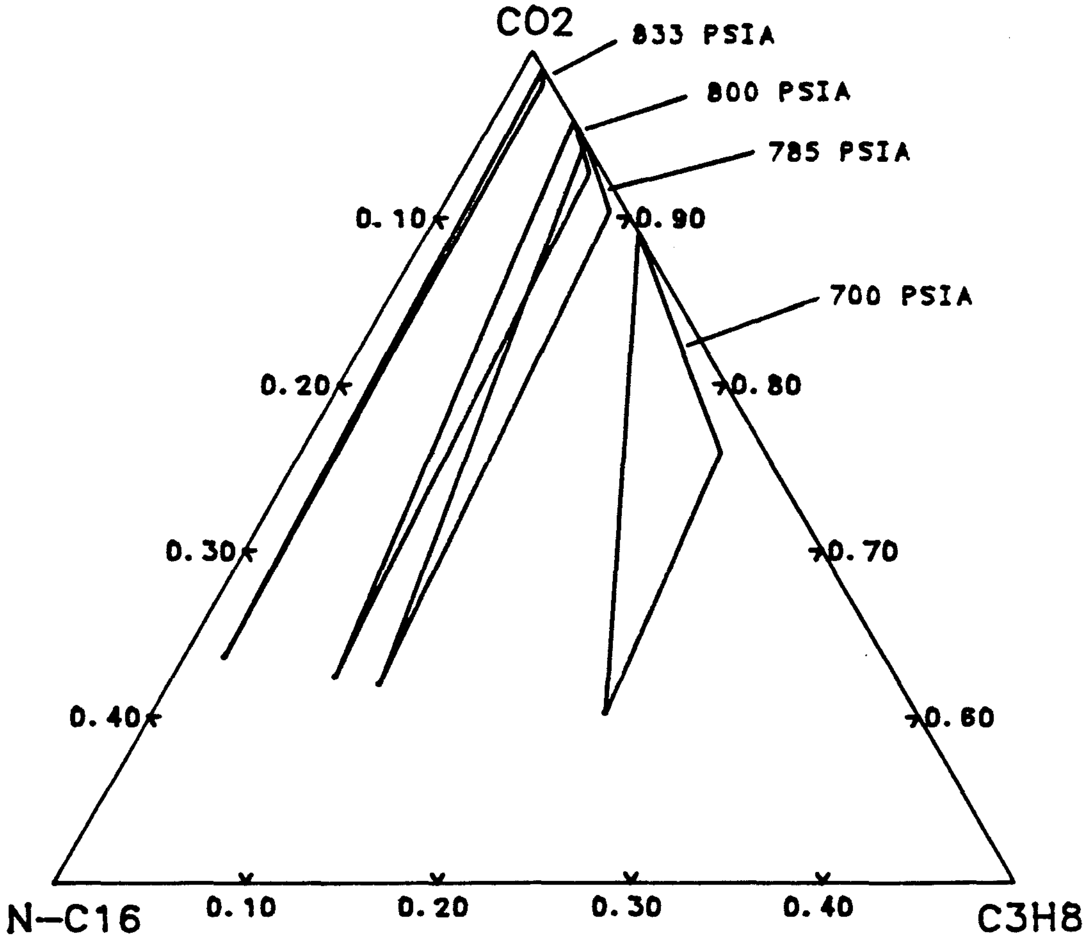


FIGURE 6.26 MOVEMENT OF THE COMPUTED THREE-PHASE REGION AS THE PRESSURE VARIES FOR THE MIXTURE OF CO2, C3H8, AND N-C16 AT 70°F.



therefore moving the three-phase region towards the $CO_2$-n-$C_{16}$ side of the triangle. This is just opposite to the behavior of the $CO_2$-$CH_4$-n-$C_{16}$ mixture. As Figure 6.19 illustrates, the three-phase region moves in the direction of lower $CO_2$ concentrations. Clearly, changing the light hydrocarbon component from $CH_4$ to $C_3H_8$ in these ternary mixtures alters the phase behavior substantially.

## 6.5 $CO_2$-Crude Oil Mixture

The final phase behavior example is a mixture of carbon dioxide with a recombined reservoir fluid. An analysis of several West Texas reservoir oils is given by Turek, Metcalfe, and Fishback [T2]. In their work, reservoir fluid samples were mixed with $CO_2$ in varying amounts. The resulting equilibrium phases were analyzed for compositional data. Several reported examples showed three-phase behavior. Recombined oil C2 was selected for this example because of its three-phase behavior when mixed with $CO_2$ and also because non-hydrocarbon components such as $N_2$ and $H_2S$ were present in only trace amounts. The composition of oil C2 is given in Table 6.13.

Although an extended analysis up to $C_{40+}$ was given for the crude oil, a ten component mixture was used in the example calculation. The components were $CO_2$, the saturated alkanes $C_1$ through $C_6$, and the $C_{7+}$ fraction which

## Table 6.13

Fluid composition analysis of recombined reservoir oil C2 taken from Turek et al. [T2].

| Component | Composition |
|---|---|
| $CO_2$ | 0.0001 |
| $C_1$ | 0.1834 |
| $C_2$ | 0.0778 |
| $C_3$ | 0.0791 |
| $i\text{-}C_4$ | 0.0065 |
| $n\text{-}C_4$ | 0.0474 |
| $i\text{-}C_5$ | 0.0165 |
| $n\text{-}C_5$ | 0.0178 |
| $C_6$'s | 0.0382 |
| $C_{7+}$ | 0.5332 |

| | |
|---|---|
| $C_{7+}$ Mol. Wt. | 230 |
| $C_{7+}$ Sp. Grav. | 0.8763 |

was taken as a single component. Isomers of $C_4$ and $C_5$ were separate components in the mixture. The alkane n-hexane was used in the calculation to represent the $C_6$'s component since no other information was given regarding its character.

The Riazi correlations [R3] were used to characterize the $C_{7+}$ fraction. The correlations have the general form

$$\theta = aT_B{}^b\gamma^c \qquad (6.5.1)$$

where $\theta$ is the predicted property, $T_B$ is the normal boiling point, and $\gamma$ is the specific gravity. The constants a, b, and c differ for each predicted property and values are listed in Table 6.14. The normal boiling point for the $C_{7+}$ fraction was estimated using the correlation for molecular weight and the data given in Table 6.13. To estimate the critical temperature and pressure, this calculated boiling point was used along with the specific gravity in the above correlation. These correlations are simple and easy-to-use.

The acentric factor of the $C_{7+}$ fraction was estimated using Edmister's [E1] equation:

Table 6.14

Constants used in the general property correlation, Equation 6.5.1.

| Property | a | b | c |
|---|---|---|---|
| Molecular weight | 4.5763E-5 | 2.1962 | -1.0164 |
| $T_c$, °R | 2.42787E01 | 0.58848 | 0.3596 |
| $P_c$, psia | 3.12281E09 | -2.3125 | 2.3201 |

$$\omega = \frac{3}{7} \frac{(\frac{T_B}{T_C})}{1 - (\frac{T_B}{T_C})} \log_{10} P_C(\text{atm}) - 1 \qquad (6.5.2)$$

Critical values for all components in the mixture are listed in Table 6.1.

Binary interaction coefficients for $C_{7+}$ with other components were computed using the correlation of Lawal [L1]:

$$\delta_{ij} = \frac{a}{T_{B\ell}}[\ln_e(|M_j - M_i| + 1)]^b \qquad (6.5.3)$$

where $T_{B\ell}$ is the boiling point of the lightest hydrocarbon, or non-hydrocarbon, in degrees Kelvin, $M_i$ is the molecular weight and a and b are correlation constants. Values of a and b are given in Table 6.15. This correlation was also used for i-$C_4$ and i-$C_5$ binary interaction coefficients with other components.

For this mixture, three-phase equilibrium was calculated. However, the highest temperature for which a three-phase mixture was found was at about 83°F. Turek et al. reported three-phase composition data that was taken at 94°F. To match the experimental data, the input data would have to be adjusted. Only ten components were used for this correlation. Only one component was used to

Table 6.15

Constants used in the binary interaction coefficient correlation, Equation 6.5.3.

| Interaction | a | b |
|---|---|---|
| Paraffin/paraffin | 1.688574E-03 | 5.382733 |
| $CO_2$/paraffin | 1.378162 | 2.155123 |

represent the $C_{7+}$ fraction, which comprised over 50 percent of the entire reservoir fluid. Clearly, better selection of the pseudo-components is necessary. Representing the $C_{7+}$ fraction by more than one component would be quite useful, especially in this example, where an extended analysis is available.

Figure 6.27 shows a phase distribution versus pressure diagram for a mixture of reservoir fluid with 75 mole percent $CO_2$ at 83°F. The three-phase region exists over a pressure range of about 40 psia. As with the $CO_2$-$CH_4$-n-$C_{16}$ mixture, at low pressures, a liquid and vapor phase coexist at equilibrium. As pressure increases, a second liquid forms creating three-phase equilibrium. As pressure continues to increase, the vapor phase shrinks and the second liquid phase grows until the vapor phase disappears altogether. At that point, only two liquid phases remain.

Table 6.16 lists equilibrium phase compositions and phase properties for a three-phase mixture. The temperature is about 83°F and the pressure is 1060 psia. The overall composition is for a mixture of the reservoir oil with 80 mole percent $CO_2$. From the phase compositions, it can be seen that the $CO_2$-lean liquid contains almost all of the $C_{7+}$ component. It is present in the second liquid in only a small amount and in trace amounts in the vapor.

FIGURE 6.27 COMPUTED PHASE DISTRIBUTION VERSUS PRESSURE FOR THE CO2-RESERVOIR OIL MIXTURE. THE OVERALL CO2 MOLE FRACTION IS 75%.

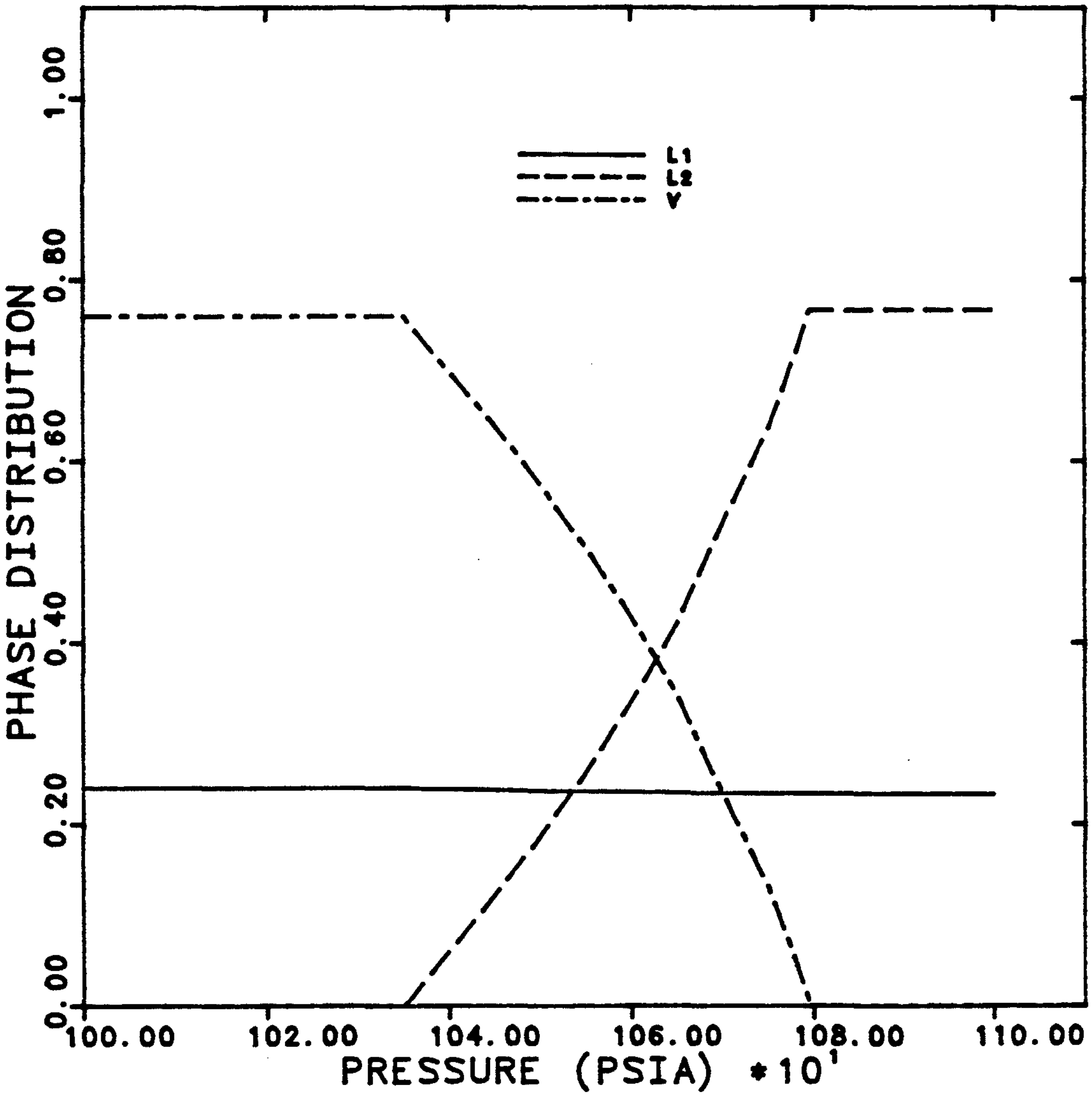


Table 6.16

Calculated phase compositions and properties for a $CO_2$ and reservoir oil mixture at 542.5°R and 1060 psia.

| | | Phase Composition | | |
|---|---|---|---|---|
| Component | Overall Composition | $L_1$ | $L_2$ | V |
| $CO_2$ | 0.80002 | 0.30228 | 0.91058 | 0.90644 |
| $CH_4$ | 0.03668 | 0.01719 | 0.03947 | 0.05158 |
| $C_2H_6$ | 0.01556 | 0.01352 | 0.01596 | 0.01633 |
| $C_3H_8$ | 0.01582 | 0.02018 | 0.01511 | 0.01306 |
| $i\text{-}C_4$ | 0.00130 | 0.00213 | 0.00115 | 0.00089 |
| $n\text{-}C_4$ | 0.00948 | 0.01734 | 0.00801 | 0.00590 |
| $i\text{-}C_5$ | 0.00330 | 0.00734 | 0.00251 | 0.00167 |
| $n\text{-}C_5$ | 0.00356 | 0.00861 | 0.00256 | 0.00162 |
| $C_6$ | 0.00764 | 0.02310 | 0.00447 | 0.00248 |
| $C_{7+}$ | 0.10664 | 0.58831 | 0.00018 | 0.00003 |
| | | Phase Properties | | |
| Phase distribution | | 0.18106 | 0.71624 | 0.10270 |
| Compressibility factor | | 0.64666 | 0.26514 | 0.38700 |
| Average molecular weight | | 154.5 | 43.2 | 42.6 |
| Density (lbs/cu ft) | | 43.50 | 29.65 | 20.06 |

The $CO_2$-rich liquid, $L_2$, is very similar in composition to the vapor phase. This is reflected in the phase densities as well. The liquid phase $L_1$ has the highest density, but it also has the largest compressibility factor of 0.64666.

## 6.6 Comparison of Numerical Methods

A comparison was made between the three different solution techniques available in the algorithm to solve the phase stability equations under several different conditions. The three techniques are the successive substitution method (SS), the secant method (SEC), and Powell's method (POW). The secant method and Powell's method are available through the IMSL Library routine which is accessible on the University of Texas CDC Dual Cyber computer system. Documentation for these two routines is available in the IMSL user's manual [I2]. The routine for Powell's method is a modification of a MINPACK [M9] subroutine and the secant method is an adaptation from Wolfe [W2].

Listed in Table 6.17 are iterations to convergence and execution times for the ten component, $CO_2$-reservoir oil mixture. The execution times given are in cpu seconds required for the CDC Dual Cyber 170/750 machine. The number of iterations required for each of the two solution vectors to converge is added together and listed in the table.

Table 6.17

Iterations and execution times for solution of phase stability equations for a ten component mixture.

| | Iterations | | | Execution time (cpu secs) | | |
|---|---|---|---|---|---|---|
| Pressure, psia | SS | SEC | POW | SS | SEC | POW |
| 500.0 | 20 | 36 | 36 | .050 | .166 | .127 |
| 800.0 | 23 | 38 | 35 | .054 | .184 | .133 |
| 900.0 | 24 | 38 | 35 | .060 | .185 | .138 |
| 1000.0 | 25 | 37 | 35 | .061 | .174 | .135 |
| 1100.0 | 32 | 38 | 37 | .077 | .183 | .144 |
| 1200.0 | 34 | 42 | 38 | .083 | .222 | .150 |
| 1500.0 | 28 | 44 | 39 | .069 | .267 | .155 |

At pressures below 1000 psia, vapor exists in equilibrium with liquid. At 1100 psia and above, there is liquid-liquid equilibrium. In all cases, the SS method takes fewer iterations and less execution time than do either the secant or Powell's method. Powell's method is always faster than the secant method.

However, in some cases, and especially near the critical point, Powell's method and the secant method are faster than the SS method. Table 6.18 lists some results for a ternary mixture of $CO_2$, $C_3H_8$, and n-$C_{16}$ at 70°F and 1000 psia for several overall compositions, all of which are in the liquid-liquid region (see Figure 6.25). As the critical point is approached, the SS method takes many more iterations than do either the secant or Powell's method. Very near the critical point, the SS method fails to converge after 3000 iterations. Due to the better convergence characteristics near the critical region, Powell's method was used for solution of the phase stability equations in all of the phase behavior examples presented above.

In Table 6.19 are given execution times needed for the solution of the phase equilibrium problem. These times include solution of the phase stability equations and also solution of the flash equations. The times listed are not for specific examples but are representative of execution times required under a variety of conditions. Near the

Table 6.18

Iterations and execution times for solution of phase stability equations for a ternary mixture.

| Overall Composition | Iterations | | | Execution time (cpu secs) | | |
|---|---|---|---|---|---|---|
| | SS | SEC | POW | SS | SEC | POW |
| .9, .05, .05 | 58 | 27 | 24 | .051 | .051 | .39 |
| .85, .10, .05 | 78 | 30 | 27 | .067 | .056 | .046 |
| .80, .15, .05 | 118 | 35 | 28 | .102 | .067 | .047 |
| .75, .20, .05 | 235 | 34 | 30 | .201 | .065 | .051 |
| .65715, .273, .06985 | DNC* | 79 | 48 | -- | .137 | .087 |

---

* Did not converge

Table 6.19

Execution times required for a phase equilibria solution, including stability analysis.

| | Execution Times (cpu sec) | |
|---|---|---|
| | Two-phase solution | Three-phase solution |
| Three component mixture | 0.18 | 0.29 |
| Ten component mixture | 0.56 | 0.64 |

critical point, for example, will require more time. The three-phase solution does not add much more execution time than that necessary for the two-phase solution since a majority of the time is spent for phase stability calculations. Both two- and three-phase solutions require about the same amount of time doing these calculations.

## 6.7 Problems Encountered with this Algorithm

Below is a brief discussion on various problems that were encountered while making the previous phase equilibria calculations.

The use of phase stability analysis as a first step in the phase equilibria algorithm worked very well. In cases where a mixture was single phase, the stability analysis detected the situation immediately without needing to make any other calculations. The successive substitution method failed to converge near critical points but worked well otherwise. By using Powell's method (or the secant method) in these difficult regions, convergence was attained relatively quickly. Phase stability analysis worked well even in cases that were very close to phase boundaries and also for cases involving liquid-liquid equilibria.

Aside from the above, only one problem was encountered using phase stability analysis. As shown in

Section 3.3, the flash calculation may converge to an unstable two-phase solution, when a stable solution is also two-phase. The phase stability analysis was unable to detect the instability of this constrained solution. By adjusting the starting solutions, however, this problem may be solved quite easily. More work needs to be done in this area. When a stable solution was actually three-phase, the corresponding constrained two-phase solution was readily detected as being unstable.

The accelerated successive substitution method for solution of the flash equations generally worked without problem. However, near phase boundaries and very near to critical points, the method failed to converge. This problem was found for all mixtures. The secant and Powell's method also had convergence problems in these areas. In particular, near the critical point, Powell's method would often converge to a trivial solution.

These difficulties encountered when computing phase diagrams could often be circumvented. For example, in a two component mixture, if the overall composition was very near a phase boundary at a given pressure, by changing the overall composition to move away from the boundary, a phase equilibrium solution could often be found at this given pressure. However, knowledge of the phase behavior at slightly lower pressures was necessary in order to determine how the overall composition should be changed. In some situations, though, changing the overall composition is not a viable alternative, and solution of these convergence problems must be made before use of the algorithm in a compositional simulator is possible.

# CHAPTER 7

## Summary and Conclusions

In the first section of this chapter is a brief summary of the work completed in this study. The second section outlines several conclusions that have been drawn from this research. The third and final section makes several recommendations for further research.

### 7.1 Summary

An algorithm for isothermal, multi-component phase equilibrium calculations has been developed for mixtures of non-reacting species. An equation of state is used to model the fluid behavior of each of up to three non-aqueous coexisting equilibrium phases. The Peng-Robinson equation of state has been used in this study.

The algorithm consists of a phase stability analysis calculation for prediction of multi-phase behavior at a given temperature, pressure, and overall composition. When necessary, as indicated by the stability analysis, a phase is added. To compute the equilibrium phase compositions and the relative amounts of each phase, a flash calculation is done. By conducting flash calculations in series with phase stability analysis, the algorithm is capable of predicting equilibria for up to three coexisting phases.

Several examples include binary and ternary mixtures of $CO_2$ with several different normal alkanes, and also a mixture of $CO_2$ with a recombined reservoir fluid. These mixtures show vapor-liquid equilibria, liquid-liquid equilibria, and also three-phase liquid-liquid-vapor behavior. Several comparisons have been made between calculated phase compostions and experimental data.

## 7.2 Conclusions

Phase stability analysis as implemented in this study was found to be an excellent method to detect multiphase behavior. Single-phase mixtures were found with relatively few calculations. The stability analysis method worked very well even in difficult to compute areas such as near the critical point and close to phase boundaries.

The successive substitution method failed to converge near the critical region but generally worked well otherwise. Therefore, methods higher in order than successive substitution are necessary to solve the stability equations in the near critical region.

Phase stability analysis combined with the flash calculation was found to be a satisfactory method which is able to predict phase equilibria for up to three equilibrium phases.

The accelerated successive substitution method for solution of the flash calculation works well in general and has convergence characteristics which are much improved over pure successive substitution. Implementation of the method is a relatively simple extension of the SS method. This is true for both the two-phase and three-phase flash calculations. Convergence problems were encountered, however, particularly, near the critical point and also near phase boundaries.

Multi-phase equilibrium calculations have been found to be very sensitive to the values used as input for the equation of state. Although the Peng-Robinson equation of state models experimental two-phase liquid-vapor behavior quite well with relatively minor adjustments of the input, three-phase behavior may be modeled quantitatively only after more substantial adjustments to the input values are made.

## 7.3 Recommendations

Further study is needed of the convergence characteristics of the flash calculation near phase boundaries and near critical points. The generation of starting values for the flash calculation from the results of the phase stability analysis should also be studied further, both from the standpoint of better initial estimates for the phase compositions, and also for the effect that the starting values have on the convergence of the flash calculation.

An effective method for elimination of extraneous phases must be found. This is crucial in cases where an unstable two-phase solution has been found from the flash calculation, but the stable solution is also two-phase.

A study should be made of the relationship between variables input into the equation of state and quantitative calculations of three-phase equilibria. This includes the role and importance of binary interaction coefficients, selection of pseudocomponents, both the number and grouping, and also the proper characterization of these pseudocomponents.

The multi-phase flash algorithm should be extended to include an aqueous phase in addition to the three non-aqueous phases already possible.

The phase-behavior package should then be installed in a compositional simulator so that the study of miscible gas processes includes the effects of the complex phase behavior between carbon dioxide and hydrocarbons.

# NOMENCLATURE

$a_i$ = temperature dependent parameter in Peng-Robinson equation of state for component i

$b_i$ = constant in Peng-Robinson equation of state for component i

$f_i$ = fugacity of component i in phase j

G = molar Gibbs free energy

$K_{ij}$ = equilibrium ratio for component i in phase j

$L_j$ = ratio of moles of phase j to total moles

$M_i$ = molecular weight of component i

nc = number of components present in the mixture

$n_{ij}$ = moles of component i in phase j

P = pressure

$P_{ci}$ = critical pressure of component i

R = universal gas constant

T = temperature

$T_B$ = normal boiling point

$T_{Ci}$ = critical temperature of component i

$U_i$ = independent variable used in phase stability analysis

v = molar volume

V = volume

$x_{ij}$ = mole fraction of component i in phase j

$y_i$ = mole fraction of component i

$Z$ = compressibility factor

$z_i$ = overall mole fraction of component i

$\gamma_i$ = activity coefficient of component i

$\delta_{ij}$ = binary interaction coefficient between components i and j

$\lambda$ = acceleration parameter in ACSS scheme

$\phi_i$ = fugacity coefficient of component i

$\omega_i$ = Pitzer acentric factor for component i

$\mu_i$ = chemical potential of component i

# APPENDIX A

## Derivation of Phase Stability Criteria

Following is a derivation of the criterion used in the phase stability analysis portion of the phase equilibria calculation given by Equation 4.2.

Consider an isolated system with a single phase composed of an nc-component mixture. The system is restricted such that no chemical reactions occur. Further, assume that the temperature, pressure, and component chemical potentials are uniform. Finally, let $z_1, z_2, \ldots, z_{nc}$ represent the component mole fractions of the overall composition.

The Gibbs free energy of this mixture, $G_O$, is given by

$$G_O = N \sum_{i=1}^{nc} z_i \mu_i(z) \qquad \text{(A.1)}$$

where N is the total number of moles and $\mu_i(z)$ is the chemical potential of component i in the mixture of composition z.

Now consider the reconstruction of the large system into two smaller systems, one portion containing $\varepsilon$ moles and a second portion containing $(N - \varepsilon)$ moles. That is, a small membrane may be placed within the system such that it encloses a finite element $\varepsilon$, now distinguishable from the rest of the system. The membrane is non-rigid, diathermal. and permeable to all components [M8]. Hence, in the original state, both the $\varepsilon$ and $(N - \varepsilon)$ portions have identical properties.

The Gibbs free energy of this change at constant temperature and pressure is given by

$$\Delta G = G_I + G_{II} - G_0 \tag{A.2}$$

where $G_I$ is the Gibbs free energy of the portion $(N - \varepsilon)$ and $G_{II}$ is the Gibbs free energy of the portion $\varepsilon$.

The Gibbs free energy is a function of temperature, pressure, and component mole numbers. But since the process is constrained to constant temperature and pressure, only the relationship with the mole numbers need be considered.

By inserting the membrane, removed from the large system is $\delta n_i$ moles of each component, for $i = 1, 2, \ldots,$ nc, with each $\delta n_i$ being independent. Thus, remaining outside the membrane are $n_1 - \delta n_1$, $n_2 - \delta n_2$, and so on, moles of each component. Hence, the following may be written:

$$G_I = G_I(n_1 - \delta n_1, n_2 - \delta n_2, \ldots, n_{nc} - \delta n_{nc}) \qquad (A.3)$$

not considering the dependence on T and P. Since each $\delta n_i$ is small, $G_I$ may be expanded about $G_0$ of the original system in a Taylor series of nc independent variables:

$$G_I = G_0(n_1, n_2, \ldots, n_{nc}) - \delta n_1 \left(\frac{\partial G}{\partial n_1}\right)_{\substack{n_j \\ j \neq 1}} - \delta n_2 \left(\frac{\delta G}{\partial n_2}\right)_{\substack{n_j \\ j \neq 2}} - \ldots - \delta n_{nc} \left(\frac{\partial G}{\partial n_{nc}}\right)_{\substack{n_j \\ j \neq nc}} + \ldots \qquad (A.4)$$

Ignoring terms of $\delta n_i{}^2$ and higher,

$$G_I = G_0 - \sum_{i=1}^{nc} \delta n_i \left(\frac{\partial G}{\partial n_i}\right)_{\substack{n_j \\ j \neq i}} \qquad (A.5)$$

Define now the set of variables $(y_1, y_2, \ldots, y_{nc})$ which represent the mole fraction of each component in the small portion $\varepsilon$. Namely,

$$y_i = \frac{\delta n_i}{\sum\limits_{k=1}^{nc} \delta n_k} = \frac{\delta n_i}{\varepsilon}$$

$$\text{for} \quad i = 1,\ 2,\ \ldots,\ nc \qquad \text{(A.6)}$$

Thus, from Equation A.5,

$$G_I = G_O - \varepsilon \sum_{i=1}^{nc} y_i \left(\frac{\partial G}{\partial n_i}\right)_{\substack{n_j \\ j \neq i}} \qquad \text{(A.7)}$$

Since the Taylor series expansion was about the original state, the partial derivatives in Equation A.7 must be evaluated at the original composition. That is to say,

$$\left(\frac{\partial G}{\partial n_i}\right)_{\substack{n_j \\ n \neq i}} = \mu_i(z) \qquad \text{A.8)}$$

Next, the Gibbs free energy of the portion $\varepsilon$ is given by

$$G_{II} = \varepsilon \sum_{i=1}^{nc} y_i \mu_i(y) \qquad \text{(A.9)}$$

Substituting Equations A.7 and A.9 into Equation A.2,

$$\Delta G = \varepsilon \sum_{i=1}^{nc} y_i\big(\mu_i(y) - \mu_i(z)\big) \qquad (A.10)$$

Stability of the original mixture requires that the Gibbs free energy be at a global minimum, or

$$\Delta G = G_I + G_{II} - G_0 \geqq 0 \qquad (A.11)$$

From Equation A.10, it follows that

$$\varepsilon \sum_{i=1}^{nc} y_i\big(\mu_i(y) - \mu_i(z)\big) \geqq 0 \qquad (A.12)$$

for all compositions y. Since $\varepsilon > 0$, a necessary criterion for stability as a single phase is that

$$\sum_{i=1}^{nc} y_i\big(\mu_i(y) - \mu_i(z)\big) \geqq 0 \qquad (A.13)$$

for all trial compositions y.

# APPENDIX B

## Listing of Source Code

```
      PROGRAM THPF (INPUT,OUTPUT,NOTE,TAPE5=INPUT,TAPE6=OUTPUT,
     1             TAPE7=NOTE)
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      DIMENSION OMFR(20), AMF1(20), AMF2(20), AMFR(20), WK1(20)

      CALL RUNDAT
      IFLASH = 1
   10 CALL DATRD (IFLASH,NC,OMFR,IPARM,WK1)
      CALL FLASH  (NC,OMFR,AMF1,AMF2,AMFR,NPH,IER)
      IF (IER.EQ.0) CALL PDENS (NPH,NC,OMFR,AMF1,AMF2,AMFR,WK1)
      CALL WRTOUT (NC,IER,NPH,OMFR,AMF1,AMF2,AMFR)
      CALL CONT (IFLASH)
      IF (IFLASH.GE.1) GO TO 10

      STOP
      END
      SUBROUTINE RUNDAT
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      COMMON /BBBB1/ ICMPN(20), IRUN1(8), IRUN2(8)

      READ (5,900) (IRUN1(I), I = 1, 8)
      READ (5,900) (IRUN2(I), I = 1,8)
      READ (5,*) IEOS, IMODE, ISA, IFC, IPL

      IF (ISA.EQ.1) THEN

             READ (5,*) EPSSA, NITSA
             EPS(1) = EPSSA

          ELSE

             READ (5,*) NDSA, NITSA
             IPARM(7) = NDSA

          ENDIF

      READ (5,*) EPSPD, NITPD, EPSTS, EPSTP
      IF (IFC.EQ.1) THEN

             READ (5,*) EPSEQR, NITEQR
```

```
         EPS(3) = EPSEQR

      ELSE

         READ (5,*) NDEQR, NITEQR, IIV
         IPARM(10) = NDEQR
         IPARM(11) =IIV

      ENDIF

  IF (IFC.LT.0) THEN

         READ (5,*) EPSEQR, EPS1, EPS2, EPS3
         EPS(3) = EPSEQR
         EPS(4) = EPS1
         EPS(5) = EPS2
         EPS(6) = EPS3

      ENDIF

  WRITE (6,100) (IRUN1(I), I = 1, 8)
  WRITE (6,110) (IRUN2(I), I = 1, 8)

  IF (IEOS.EQ.1) THEN
        WRITE (6,120) IMODE
     ELSE
        WRITE (6,130) IMODE
     ENDIF

  WRITE (6,140)
  IF (ISA.EQ.1) THEN

         WRITE (6,150)
         WRITE (6,160) EPSSA, NITSA

      ELSE

         IF (ISA.EQ.2) THEN

                WRITE (6,170)
                WRITE (6,180) NDSA, NITSA

             ELSE

                WRITE (6,190)
                WRITE (6,180) NDSA, NITSA

             ENDIF

       ENDIF
```

```
   WRITE (6,200) EPSPD, NITPD
   IF (IFC.LT.0) GO TO 10

   IF (IFC.EQ.1) THEN

          WRITE (6,210)
          WRITE (6,160) EPSEQR, NITEQR

       ELSE

          IF (IFC.EQ.2) THEN

                 WRITE (6,170)
                 WRITE (6,180) NDEQR, NITEQR

              ELSE

                 WRITE (6,190)
                 WRITE (6,180) NDEQR, NITEQR

              ENDIF

       ENDIF

   GO TO 20

10 IF (IFC.EQ.-2) THEN

          WRITE (6,220)
          WRITE (6,180) NDEQR, NITEQR
          WRITE (6,240) EPS1, EPS2, EPS3

       ELSE

          WRITE (6,230)
          WRITE (6,180) NDEQR, NITEQR
          WRITE (6,240) EPS1, EPS2, EPS3

       ENDIF

20 IF (IPL.EQ.0) GO TO 30
   WRITE (7,100) (IRUN1(I), I = 1, 8)
   WRITE (7,110) (IRUN2(I), I = 1, 8)
   IF (IFC.NE.1) THEN
          WRITE (7,910) IIV
       ENDIF

30 IPARM(1) = IEOS
   IPARM(2) = IMODE
   IPARM(3) = ISA
   IPARM(4) = IFC
```

```
      IPARM(5) = IPL
      IPARM(6) = NITSA
      IPARM(8) = NITPD
      IPARM(9) = NITEQR
      EPS(2) = EPSPD
      EPS(7) = EPSTS
      EPS(8) = EPSTP

      RETURN

  100 FORMAT (1H1,1X,8A10)
  110 FORMAT (1X,8A10)
  120 FORMAT (/,1X,*MODE OF OPERATION:*,I5,6H PHASE,
     1   /,1X,*CALCULATIONS USING PENG-ROBINSON EOS*)
  130 FORMAT (/,1X,*MODE OF OPERATION:*,I5,6H PHASE,
     1   /,1X,*CALCULATIONS USING SOAVE-REDLICH KWONG EOS*)
  140 FORMAT (/,1X,*PHASE STABILITY ANALYSIS CALCULATION*)
  150 FORMAT (/,6X,*METHOD OF SOLUTION*,15X,2HSS)
  160 FORMAT (6X,*TOLERANCE*,11X,E15.3,
     1   /,6X,*MAXIMUM ITERATIONS*,7X,I10)
  170 FORMAT (/,6X,*METHOD OF SOLUTION*,11X,6HSECANT)
  180 FORMAT (6X,22HTOLERANCE (SIG DIGITS) ,I13,
     1   /,6X,*MAXIMUM ITERATIONS*,7X,I10)
  190 FORMAT (/,6X,*METHOD OF SOLUTION*,11X,6HPOWELL)
  200 FORMAT (/,1X,*PHASE MOLE DISTRIBUTION CALCULATION*,
     1   //,6X,*TOLERANCE*,11X,E15.3,
     2   /,6X,*MAXIMUM ITERATIONS*,7X,I10,
     3   //,1X,*FLASH CALCULATION*)
  210 FORMAT (/,6X,*METHOD OF SOLUTION*,13X,4HACSS)
  220 FORMAT (/,6X,*METHOD OF SOLUTION*,6X,*ACSS/SECANT*)
  230 FORMAT (/,6X,*METHOD OF SOLUTION*,6X,*ACSS/POWELL*)
  240 FORMAT (/,6X,*SWITCHING TOLERANCES*,
     1   /,11X,*TOL 1*,10X,E15.3,/,11X,*TOL 2*,10X,E15.3,
     2   /,11X,*TOL 3*,10X,E15.3)
  900 FORMAT (8A10)
  910 FORMAT (//,1X,*INDEPENDENT VARIABLE FLAG*,10X,I5)

      END
      SUBROUTINE DATRD (IFLASH,NC,OMFR,IPARM,ALPHAI)
      COMMON /AAAA3/ TEM, PRS
      COMMON /AAAA4/ TEMC(20), PRSC(20), ACF(20), AMW(20)
      COMMON /AAAA5/ AI(20), BI(20), AIJ(20,20)
      COMMON /BBBB1/ ICMPN(20), IRUN1(8), IRUN2(8)
      COMMON /BBBB4/ API(20), AMI(20), BICM(20,20)
      DIMENSION OMFR(1), IPARM(1), ALPHAI(1)

      IF (IFLASH.GT.1) GO TO 40
      R = 10.732
      IF (IPARM(1).EQ.1) THEN

             CAI = 0.457235529 * R * R
```

```
            CBI = 0.077796074 * R
            CM0 = 0.37464
            CM1 = 1.54226
            CM2 = 0.26992

         ELSE

            CAI  = 0.4274802 * R * R
            CBI = 0.08664035 * R
            CM0 = 0.480
            CM1 = 1.574
            CM2 = 0.176

         ENDIF

      READ (5,*) NC
      DO 10, I = 1, NC

         READ (5,900) ICMPN(I), TEMC(I), PRSC(I), ACF(I), AMW(I)
         BI(I) = CBI * TEMC(I) / PRSC(I)
         API(I) = CAI * TEMC(I)**2 / PRSC(I)
         AMI(I) = CM0 + ACF(I) * (CM1 - CM2 * ACF(I))

   10 CONTINUE

      READ (5,*) ((BICM(I,J), J = 1, NC), I = 1, NC)
      DO 30, I = 1, NC-1
         DO 20, J = I+1, NC
            BICM(J,I) = BICM(I,J)
   20    CONTINUE
   30 CONTINUE

   40 IF (IFLASH.GT.2) GO TO 80

      READ (5,*) TEM
      DO 50, I = 1, NC

         ALPHAI(I) = (1.0 + AMI(I) * (1.0 - (TEM / TEMC(I))**0.5))**2
         AI(I) = API(I) * ALPHAI(I)

   50 CONTINUE

      DO 70, I = 1, NC
         DO 60, J = I, NC

            AIJ(I,J) = (1.0 - BICM(I,J)) * (AI(I) * AI(J))**0.5
            AIJ(J,I) = AIJ(I,J)

   60    CONTINUE
   70 CONTINUE
```

```
      IF (IFLASH.EQ.2) RETURN

   80 IF (IFLASH.GT.3) GO TO 90

      READ  (5,*) PRS
      IF (IFLASH.EQ.3) RETURN

   90 IF (IFLASH.GE.5) CALL DEFLT

      READ (5,*) (OMFR(I), I = 1, NC)

      IF (IFLASH.EQ.4) RETURN

      WRITE (6,200)
      WRITE (6,210) (ICMPN(I), TEMC(I), PRSC(I), ACF(I),
     1               AMW(I), I = 1, NC)

      WRITE (6,220)
      DO 100, I = 1, NC
         WRITE (6,230) ICMPN(I), I
  100 CONTINUE

      WRITE (6,240) (I, I = 1, NC)
      DO 110, I = 1, NC
         WRITE (6,250) I, (BICM(I,J), J = 1, NC)
  110 CONTINUE

      WRITE (6,260)

      RETURN

  200 FORMAT (/,14(2H -),*SYSTEM DEFINITION*,14(2H- ),
     1   //,1X,10X,2(7X,8HCRITICAL),7X,8HACENTRIC,6X,9HMOLECULAR,
     2   /,2X,9HCOMPONENT,4X,11HTEMPERATURE,7X,8HPRESSURE,
     3   9X,6HFACTOR,9X,6HWEIGHT,/,19X,7H(DEG R),9X,6H(PSIA),
     4   /,1X,10H--------- ,4(4X,11H---------- ),/)
  210 FORMAT (1X,A10,4F15.3)
  220 FORMAT (//,1X,*BINARY INTERACTION PARAMETERS*,
     1   //,6X,*DEFINE:*,)
  230 FORMAT (1X,A10,1X,1H=,I4)
  240 FORMAT (//,3X,3HJ =,20I6,/)
  250 FORMAT (1X,3HI =,I3,20F6.3)
  260 FORMAT (/,14(2H -),*END OF SYSTEM DATA*,14(2H-  ))
  900 FORMAT (A10,4F10.3)

      END
      SUBROUTINE PDENS (NPH,NC,OMFR,AMF1,AMF2,AMFR,WK1)
      COMMON /AAAA4/ TEMC(20), PRSC(20), ACF(20), AMW(20)
      COMMON /BBBB2/ FGC1(20), FGC2(20), FGCR(20), Z1, Z2, ZR
      COMMON /BBBB3/ AMW1, AMW2, AMWR, DEN1, DEN2, DENR
      DIMENSION OMFR(1), AMF1(1), AMF2(1), AMFR(1), WK1(1)
```

```
      IF (NPH.GE.2) GO TO 10

      CALL FGCF (NC,OMFR,FGCR,WK1,ZR)
      CALL DENS (NC,AMW,OMFR,ZR,AMWR,DENR)
      RETURN

   10 CALL FGCF (NC,AMF1,FGC1,WK1,Z1)
      CALL DENS (NC,AMW,AMF1,Z1,AMW1,DEN1)
      CALL FGCF (NC,AMFR,FGCR,WK1,ZR)
      CALL DENS (NC,AMW,AMFR,ZR,AMWR,DENR)
      IF (NPH.EQ.3) GO TO 20

      RETURN

   20 CALL FGCF (NC,AMF2,FGC2,WK1,Z2)
      CALL DENS (NC,AMW,AMF2,Z2,AMW2,DEN2)

      RETURN
      END
      SUBROUTINE DENS (NC,AMW,AMFR,Z,PAMW,DENSP)
      COMMON /AAAA3/ TEM, PRS
      DIMENSION AMW(1), AMFR(1)

      R = 10.73
      PAMW = 0.0
      DO 10, I = 1, NC
         PAMW = PAMW +  AMW(I) * AMFR(I)
   10 CONTINUE

      C = PRS / (R * TEM)
      DENSP = PAMW * C / Z

      RETURN
      END
      SUBROUTINE WRTOUT (NC,IER,NPH,OMFR,AMF1,AMF2,AMFR)
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      COMMON /AAAA3/ TEM, PRS
      COMMON /AAAA6/ TIME(9)
      COMMON /BBBB1/ ICMPN(20), IRUN1(8), IRUN2(8)
      COMMON /BBBB2/ FGC1(20), FGC2(20), FGCR(20), Z1, Z2, ZR
      COMMON /BBBB3/ AMW1, AMW2, AMWR, DEN1, DEN2, DENR
      DIMENSION OMFR(1), AMF1(1), AMF2(1), AMFR(1)

      IF (IER.NE.0) GO TO 30
      WRITE (6,100)
      WRITE (6,110) TEM, PRS
      IF (NPH.GT.1) GO TO 10

      AMOLR = 1.0
      WRITE (6,120) ZR, AMOLR, AMWR, DENR
```

```
      WRITE (6,130)
      WRITE (6,140) (ICMPN(I), OMFR(I), FGCR(I), I = 1, NC)
      GO TO 20

   10 IF (NPH.EQ.2) THEN

             WRITE (6,150)
             WRITE (6,160) Z1, ZR, AMOL1, AMOLR
             WRITE (6,170) AMW1, AMWR, DEN1, DENR
             WRITE (6,180)
             WRITE (6,190) (ICMPN(I), OMFR(I), AMF1(I), AMFR(I),
     1                      FGC1(I), FGCR(I), I = 1, NC)

          ELSE

             WRITE (6,200)
             WRITE (6,210) Z1, Z2, ZR, AMOL1, AMOL2, AMOLR
             WRITE (6,220) AMW1, AMW2, AMWR, DEN1, DEN2, DENR
             WRITE (6,230)
             WRITE (6,240) (ICMPN(I), OMFR(I), AMF1(I), AMF2(I),
     1        AMFR(I), FGC1(I), FGC2(I), FGCR(I), I = 1, NC)

          ENDIF

   20 WRITE (6,250)
      IF (IPARM(3).EQ.1) THEN
             WRITE (6,260) IPARM(12), IPARM(13)
          ELSE
             WRITE (6,270) IPARM(14), IPARM(15)
          ENDIF

      T = TIME(3) - TIME(2)
      WRITE (6,280) T

      IF (NPH.GE.2) THEN

             WRITE (6,290)
             IF (IPARM(4).LE.1) WRITE (6,300) IPARM(16)
             IF (IABS(IPARM(4)).GE.2) WRITE (6,310) IPARM(17)
             T = TIME(5) - TIME(4)
             WRITE (6,280) T

          ENDIF

      IF (NPH.EQ.3) THEN

             ITIT = IPARM(19) + IPARM(20) + IPARM(21) + IPARM(22)
             T = TIME(7) - TIME(6)
             WRITE (6,320) ITIT
             WRITE (6,280) T
             WRITE (6,330)
```

```
        IF (IPARM(4).LE.1) WRITE (6,300) IPARM(23)
        IF (IABS(IPARM(4)).GE.2) WRITE (6,310) IPARM(24)
        T = TIME(9) - TIME(8)
        WRITE (6,280) T

      ENDIF

    T = TIME(9) - TIME(1)
    WRITE (6,340) T
    IF (NPH.GE.2) WRITE (6,350) IPARM(18), RPARM(3)
    IF (NPH.EQ.3) WRITE (6,360) IPARM(25), RPARM(5)
    IF (IPARM(26).NE.0) WRITE (6,370) IPARM(23)
    GO TO 40

 30 CALL ERROPR (IER)

 40 WRITE (6,380)

    RETURN

100 FORMAT (/,13(2H -),*EQUILIBRIUM SOLUTION*,14(2H- ))
110 FORMAT (//,1X,*SYSTEM TEMPERATURE (DEG R)*,5X,F12.4,
  1   /,1X,*SYSTEM PRESSURE (PSIA)*,9X,F12.4)
120 FORMAT (//,1X,*SINGLE PHASE SYSTEM*,//,6X,*COMPRESSIBILITY*,/,
  1   15X,6HFACTOR,F15.5,/,1X,*RATIO OF PHASE MOLES*,
  2   /,7X,14HTO TOTAL MOLES,F15.5,/,4X,*AVERAGE MOLECULAR*,
  3   /,15X,6HWEIGHT,F15.2,/,9X,*MASS DENSITY*,/,
  4   8X,13H(LBS / CU FT),F15.2)
130 FORMAT (//,2X,9HCOMPONENT,7X,13HMOLE FRACTION,
  1   5X,20HFUGACITY COEFFICIENT,/,1X,10H---------- ,
  2   5X,15H--------------- ,5X,20(1H-),/)
140 FORMAT (1X,A10,E20.10,E25.10)
150 FORMAT (/,1X,*TWO PHASE SYSTEM*,//,1X,28X,7HPHASE 1,
  1   8X,7HPHASE 2,/,21X,2(8X,7H------- ),/)
160 FORMAT (6X,*COMPRESSIBILITY*,/,15X,6HFACTOR,2F15.5,
  1   /,1X,*RATIO OF PHASE MOLES*,/,7X,14HTO TOTAL MOLES,2F15.5)
170 FORMAT (4X,*AVERAGE MOLECULAR*,/,15X,6HWEIGHT,2F15.2,
  1   /,9X,*MASS DENSITY*,/,8X,13H(LBS / CU FT) ,2F15.2)
180 FORMAT (//,1X,44X,*PHASE COMPOSITION*,/,24X,7HOVERALL,15X,
  1   15H(MOLE FRACTION),23X,*FUGACITY COEFFICIENT*,
  2   /,27X,4HMOLE,5X,35(1H-),5X,35(1H-),/,
  3   2X,9HCOMPONENT,12X,8HFRACTION,2(13X,7HPHASE 1,13X,7HPHASE 2),
  4   /,1X,10H---------- ,5(5X,15H--------------- ),/)
190 FORMAT (1X,A10,5E20.10)
200 FORMAT (/,1X,*THREE PHASE SYSTEM*,//,1X,28X,7HPHASE 1,
  1   8X,7HPHASE 2,8X,7HPHASE 3,/,21X,3(8X,7H------- ),/)
210 FORMAT (6X,*COMPRESSIBILITY*,/,15X,6HFACTOR,3F15.5,
  1   /,1X,*RATIO OF PHASE MOLES*,/,7X,14HTO TOTAL MOLES,3F15.5)
220 FORMAT (4X,*AVERAGE MOLECULAR*,/,15X,6HWEIGHT,3F15.2,
  1   /,9X,*MASS DENSITY*,/,8X,13H(LBS / CU FT),3F15.2)
230 FORMAT (//,1X,42X,*PHASE COMPOSITION*,/,19X,7HOVERALL,18X,
```

```
     1   15H(MOLE FRACTION),27X,*FUGACITY COEFFICIENT*,
     2   /,22X,4HMOLE,5X,40(1H-),5X,40(1H-),/,
     3   2X,9HCOMPONENT,7X,8HFRACTION,2(8X,7HPHASE 1,8X,7HPHASE 2,
     4   8X,7HPHASE 3),/,1X,10(1H-),7(5X,10H---------- ),/)
 240 FORMAT (1X,A10,7E15.8)
 250 FORMAT (//,1X,*PHASE STABILITY ANALYSIS*,/)
 260 FORMAT (6X,*NUMBER OF ITERATIONS*,12X,I3,2H /,I3)
 270 FORMAT (6X,*NUMBER OF FUNCTION EVALUATIONS*,2X,I3,2H /,I3)
 280 FORMAT (6X,*EXECUTION TIME (CPU SECS)*,5X,F10.4)
 290 FORMAT (/,1X,*TWO PHASE SPLIT CALCULATION*,/)
 300 FORMAT (6X,*NUMBER OF ITERATIONS (ACSS)*,3X,I10)
 310 FORMAT (6X,*NUMBER OF FUNCTION EVALUATIONS*,I10)
 320 FORMAT (/,1X,*MULTI-PHASE STABILITY TEST*,/,
     1   6X,*ITERATIONS (FUNCTION EVALUATIONS)*,I7)
 330 FORMAT (/,1X,*THREE PHASE SPLIT CALCULATION*,/)
 340 FORMAT (/,1X,*FLASH COMPUTATION*,//,6X,
     1   *TOTAL EXECUTION TIME (CPU SECS)*,F9.4)
 350 FORMAT (6X,*TOTAL ITERATIONS OF TWO PHASE*,/,11X,*DISTRIBUTION*,
     1   13X,I10,/,6X,*TWO PHASE VECTOR NORM*,E19.7)
 360 FORMAT (6X,*TOTAL ITERATIONS OF THREE*,
     1   /,11X,*PHASE DISTRIBUTION*,7X,I10,
     2   /,6X,*THREE PHASE VECTOR NORM*,E17.10)
 370 FORMAT (/,1X,*NOTE:  THIRD PHASE ELIMINATED AFTER*,
     1   I5,1X,*ITERATIONS*)
 380 FORMAT (/,14(2H -),*END OF RESULTS*,14(2H- ))

     END
     SUBROUTINE CONT (IFLASH)

     READ (5,*) IFLASH
     IF (IFLASH.LE.0) GO TO 10
     RETURN

  10 WRITE (6,100)
     RETURN

 100 FORMAT (//,1X,*END OF PROGRAM*)

     END
     SUBROUTINE DEFLT
     WRITE (6,100)
     STOP
 100 FORMAT (/,1X,*AN ERROR HAS BEEN MADE IN THE VALUE OF IFLASH*)
     END
     SUBROUTINE ERROPR (IER)
     COMMON /AAAA3/ TEM, PRS

     IF (IER.GT.200) GO TO 10
     IF (IER.EQ.111) WRITE (6,100)
     IF (IER.EQ.112) WRITE (6,110)
     IF (IER.EQ.121) WRITE (6,120)
```

```
      IF (IER.EQ.122) WRITE (6,130)
      IF (IER.EQ.131) WRITE (6,140)
      IF (IER.EQ.132) WRITE (6,150)
      GO TO 40

   10 IF (IER.GT.300) GO TO 30
      IF (IER.EQ.210) WRITE (6,160)
      IF (IER.EQ.220) WRITE (6,170)
      IF (IER.EQ.230) WRITE (6,180)
      GO TO 40

   30 IF (IER.EQ.301) WRITE (6,190)
      IF (IER.EQ.401) WRITE (6,200)
      IF (IER.EQ.402) WRITE (6,210)
      IF (IER.EQ.403) WRITE (6,220)
      IF (IER.EQ.800) WRITE (6,230)

   40 WRITE (6,900) TEM, PRS

      STOP

  100 FORMAT (/,1X,*ERROR IN SS OF SA IN SOL 1*,/)
  110 FORMAT (/,1X,*ERROR IN SS OF SA IN SOL 2*,/)
  120 FORMAT (/,1X,*ERROR IN SECANT OF SA IN SOL 1*,/)
  130 FORMAT (/,1X,*ERROR IN SECANT OF SA IN SOL 2*,/)
  140 FORMAT (/,1X,*ERROR IN POWELL OF SA IN SOL 1*,/)
  150 FORMAT (/,1X,*ERROR IN POWELL OF SA IN SOL 2*,/)
  160 FORMAT (/,1X,*ERROR IN ACSS OF PHASE SPLIT*,/)
  170 FORMAT (/,1X,*ERROR IN SECANT OF PHASE SPLIT*,/)
  180 FORMAT (/,1X,*ERROR IN POWELL OF PHASE SPLIT*,/)
  190 FORMAT (/,1X,*MOLE NUMBER IS NEGATIVE IN PHASE DIST.*)
  200 FORMAT (/,1X,*NON CON. IN FINDING PHASE DIST.*)
  210 FORMAT (/,1X,*DERIVATIVE IS ZERO IN PHASE DIST.*)
  220 FORMAT (/,1X,*MOLE DISTRIBUTION OUT OF RANGE*)
  230 FORMAT (/,1X,*ALGORITHM UNABLE TO FIND STABLE SOLUTION*)
  900 FORMAT (/,1X,*PROBLEM OCCURRED AT:*,/,
     1   1X,*SYSTEM TEMPERATURE (DEG R)*,F17.4,
     2   /,1X,*SYSTEM PRESSURE (PSIA)*,F17.4)

      END
      SUBROUTINE FLASH (NC,OMFR,AMF1,AMF2,AMFR,NPH,IER)
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      COMMON /AAAA6/ TIME(9)
      DIMENSION OMFR(1), AMF1(1), AMF2(1), AMFR(1)
      DIMENSION WRK1(20), WRK2(20), WK(1420)

      CALL SECOND (TIME(1))
      DO 10, I = 12, 26
         IPARM(I) = 0
   10 CONTINUE
```

```
      IPL = IPARM(5)
      CALL SECOND(TIME(2))
      CALL PSA (NC,OMFR,WRK1,AMF1,AMFR,WK,WRK2,IER)
      IF (IER.NE.0) GO TO 60
      CALL SECOND (TIME(3))
      IF (IPL.GE.1) CALL WRT2 (NC,OMFR,AMF1,AMFR)
      CALL SOLAN (NC,AMF1,AMFR,OMFR,EPS,NPH)
      IF (NPH.EQ.2) GO TO 20
      CALL SECOND (TIME(9))
      RETURN

   20 CALL SECOND (TIME(4))
   30 CALL PSPLT (NC,OMFR,AMF1,AMFR,WRK1,WK,IER)
      IF (IER.NE.0) GO TO 60
      CALL SECOND (TIME(5))
      IF (IPARM(2).EQ.3) GO TO 40
      CALL SECOND (TIME(9))
      RETURN

   40 CALL WRT5(NC,AMF1,AMFR)
      CALL SECOND (TIME(6))
      CALL PSATP (NC,AMF1,AMF2,AMFR,WRK1,WK,NPH,IER)
      IF (IER.NE.0) GO TO 60
      CALL SECOND (TIME(7))
      CALL SECOND (TIME(9))
      IF ((NPH.EQ.3).AND.(IPARM(26).EQ.0)) GO TO 50
      IF (NPH.EQ.3) IER = 800
      RETURN

   50 CALL SECOND (TIME(8))
      CALL PSPLT3 (NC,OMFR,AMF1,AMF2,AMFR,WRK1,WRK2,IER)
      IF (IPARM(26).NE.0) GO TO 30
      CALL SECOND (TIME(9))

   60 RETURN
      END
      SUBROUTINE PSA (NC,OMFR,HG,AMF1,AMF2,WK,WK1,IER)
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      DIMENSION OMFR(1), HG(1), AMF1(1), AMF2(1)
      DIMENSION WK(1), WK1(1)

      IPL = IPARM(5)
      IMET = IPARM(3)

      CALL SETH (NC,OMFR,HG,WK1)
      CALL SETY (NC,OMFR,AMF1,AMF2,WK1)
      IF (IPL.GE.1) CALL WRT1 (NC,WK1,AMF1,AMF2)
      IF (IMET.GE.2) GO TO 10
      CALL PSASS (NC,HG,AMF1,AMF2,IER)
      GO TO 20
```

```
10 CALL PSAHOM (NC,OMFR,HG,AMF1,AMF2,WK,WK1,IER)

20 RETURN
   END
   SUBROUTINE SETH (NC,OMFR,HG,WK)
   DIMENSION OMFR(1), HG(1), WK(1)

   CALL FGCF (NC,OMFR,HG,WK,Z)
   DO 10, I = 1, NC
      HG(I) = ALOG(OMFR(I) * HG(I))
10 CONTINUE

   RETURN
   END
   SUBROUTINE SETY (NC,OMFR,AMF1,AMF2,EQR)
   COMMON /AAAA2/ IWK, AMN(20), WK2(20), WK3(20), WK4(20), WK5(20)
   COMMON /AAAA3/ TEM, PRS
   COMMON /AAAA4/ TEMC(20), PRSC(20), ACF(20), AMW(20)
   DIMENSION OMFR(1), AMF1(1), AMF2(1), EQR(1)

   DO 10, I = 1, NC

      A = 5.37 * (1.0 + ACF(I) - TEMC(I) *  (1.0 + ACF(I)) / TEM)
      EQR(I) = PRSC(I) * EXP(A) / PRS
      AMN(I) = EQR(I) * OMFR(I)

10 CONTINUE

   CALL MFNM (NC,AMN,AMF1)

   DO 20, I = 1, NC
      AMN(I) = OMFR(I) / EQR(I)
20 CONTINUE
   CALL MFNM (NC,AMN,AMF2)

   ZM1 = 0.0
   ZM2 = 0.0
   DO 30, I = 1, NC

      IF (AMF1(I).GE.ZM1) THEN

             ZM1 = AMF1(I)
             I1 = I

          ENDIF

      IF (AMF2(I).GE.ZM2) THEN

             ZM2 = AMF2(I)
             I2 = I
```

```
              ENDIF

   30 CONTINUE

      ICY = 0
      IF (I1.EQ.I2) THEN

              C1 = AMF1(I1) - OMFR(I1)
              C2 = AMF2(I2) - OMFR(I2)
              CSS = C1 * C2
              IF (CSS.GE.0.0) ICY = 1

          ENDIF

      IF (ICY.EQ.0) GO TO 50

      R = OMFR(I1) / (1.0 - OMFR(I1))
      IF (ABS(C1).GT.ABS(C2)) THEN

              IF (AMF1(I1).LT.OMFR(I1)) R = 1.0 / R
              ADMF = OMFR(I1) - R * C1
              RMF = (1.0 - ADMF) / FLOAT(NC-1)
              DO 41, I = 1, NC
                 AMF2(I) = RMF
   41         CONTINUE
              AMF2(I1) = ADMF

          ELSE

              IF (AMF2(I1).LT.OMFR(I1)) R = 1.0 / R
              ADMF = OMFR(I1) - R * C2
              RMF = (1.0 - ADMF) / FLOAT(NC-1)
              DO 42, I = 1, NC
                 AMF1(I) = RMF
   42         CONTINUE
              AMF1(I1) = ADMF

          ENDIF

   50 RETURN
      END
      SUBROUTINE MFNM (NC,AMN,AMF)
      DIMENSION AMN(1), AMF(1)

      SUM = 0.0
      IM = 0
      ZMAX = 0.0
      DO 10, I = 1, NC

         IF (AMN(I).GE.ZMAX) THEN
               ZMAX = AMN(I)
```

```
            IM = I
         ENDIF
      SUM = SUM + AMN(I)

10 CONTINUE

   SUMF = 0.0
   DO 20, I = 1, NC

      IF (I.EQ.IM) GO TO 20
      AMF(I) = AMN(I) / SUM
      SUMF = SUMF + AMF(I)

20 CONTINUE

   AMF(IM) = 1.0 - SUMF

   RETURN
   END
   SUBROUTINE PSASS (NC,HG,AMF1,AMF2,IER)
   COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
   COMMON /AAAA2/ IWK, FGC(20), AMF(20), WK3(20), WK4(20), WK5(20)
   DIMENSION HG(1), AMF1(1), AMF2(1)

   EPSSA = EPS(1)
   MAXIT = IPARM(6)

   DO 10, I = 1, MAXIT

      NIT = I
      CALL MFNM (NC,AMF1,AMF)
      CALL FGCF (NC,AMF,FGC,WK3,Z)
      CALL CCKSA (NC,AMF1,FGC,HG,FN1)
      IF (FN1.LE.EPSSA) GO TO 20
      CALL UPD (NC,FGC,HG,AMF1)

10 CONTINUE
   IER = 111
   GO TO 40

20 IPARM(12) = NIT
   RPARM(1) = FN1
   DO 30, I = 1, MAXIT

      NIT = I
      CALL MFNM (NC,AMF2,AMF)
      CALL FGCF (NC,AMF,FGC,WK3,Z)
      CALL CCKSA (NC,AMF2,FGC,HG,FN2)
      IF (FN2.LE.EPSSA) GO TO 50
      CALL UPD (NC,FGC,HG,AMF2)
```

```
30 CONTINUE
   IER = 112
40 RETURN

50 IPARM(13) = NIT
   RPARM(2) = FN2
   IER = 0

   RETURN
   END
   SUBROUTINE UPD (NC,FGC,HG,AMF)
   DIMENSION FGC(1), HG(1), AMF(1)

   DO 10, I = 1, NC
      AMF(I) = EXP(HG(I) - ALOG(FGC(I)))
10 CONTINUE

   RETURN
   END
   SUBROUTINE CCKSA (NC,AMF,FGC,HG,FN)
   DIMENSION AMF(1), FGC(1), HG(1)

   SUM = 0.0
   DO 10, I = 1, NC

      HOY = ALOG(AMF(I) * FGC(I)) - HG(I)
      SUM = SUM + HOY**2

10 CONTINUE

   FN = SQRT(SUM)
   RETURN
   END
   SUBROUTINE PSAHOM (NC,OMFR,HG,AMF1,AMF2,WK,AIV,IER)
   COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
   COMMON /AAAA2/ NFE, WK1(20), WK2(20), WK3(20), WK4(20), WK5(20)
   DIMENSION OMFR(1), HG(1), AMF1(1), AMF2(1), WK(1), AIV(1)
   EXTERNAL HFCN

   MAXIT = IPARM(6) - IPARM(12)
   NSIG = IPARM(7)
   NFE = 0

   DO 10, I = 1, NC
      AIV(I) = ALOG(AMF1(I))
10 CONTINUE

   IF (IABS(IPARM(3)).EQ.2) THEN

         CALL ZSCNT (HFCN,NSIG,NC,MAXIT,HG,AIV,FN1,WK,IER)
         IF (IER.EQ.130) CALL SOPROB (NC,OMFR,AIV,IER)
```

```
          IF (IER.NE.0) IER = 121

       ELSE

          CALL ZSPOW (HFCN,NSIG,NC,MAXIT,HG,AIV,FN1,WK,IER)
          IF (IER.EQ.131) CALL SOPROB (NC,OMFR,AIV,IER)
          IF (IER.NE.0) IER = 131

       ENDIF
   IF (IER.NE.0) GO TO 40

   DO 20, I = 1, NC

      AMF1(I) = EXP(AIV(I))
      AIV(I) = ALOG(AMF2(I))

20 CONTINUE

   IPARM(14) = NFE
   MAXIT = IPARM(6) - IPARM(13)
   IF (IABS(IPARM(3)).EQ.2) THEN

          CALL ZSCNT (HFCN,NSIG,NC,MAXIT,HG,AIV,FN2,WK,IER)
          IF (IER.EQ.130) CALL SOPROB (NC,OMFR,AIV,IER)
          IF (IER.NE.0) IER = 122

       ELSE

          CALL ZSPOW (HFCN,NSIG,NC,MAXIT,HG,AIV,FN2,WK,IER)
          IF (IER.EQ.131) CALL SOPROB (NC,OMFR,AIV,IER)
          IF (IER.NE.0) IER = 132

       ENDIF
   IF (IER.NE.0) GO TO 40

   DO 30, I = 1, NC
      AMF2(I) = EXP(AIV(I))
30 CONTINUE

   IPARM(15) = NFE - IPARM(14)
   RPARM(1) = FN1
   RPARM(2) = FN2

40 RETURN
   END
   SUBROUTINE HFCN (AIV,HOY,NC,HG)
   COMMON /AAAA2/ NFE, AMF(20), FGC(20), AMN(20), WK4(20), WK5(20)
   DIMENSION AIV(1), HOY(1), HG(1)

   DO 10, I = 1, NC
      AMN(I) = EXP(AIV(I))
```

```
10 CONTINUE

   CALL MFNM (NC,AMN,AMF)
   CALL FGCF (NC,AMF,FGC,WK4,Z)

   DO 20, I = 1, NC
      HOY(I) = AIV(I) + ALOG(FGC(I)) - HG(I)
20 CONTINUE

   NFE = NFE + 1

   RETURN
   END
   SUBROUTINE SOPROB (NC,OMFR,AIV,IER)
   DIMENSION OMFR(1), AIV(1)

   DO 10, I = 1, NC
      AIV(I) = ALOG(OMFR(I))
10 CONTINUE

   IER = 0

   RETURN
   END
   SUBROUTINE SOLAN (NC,AMF1,AMF2,OMFR,EPS,NPH)
   COMMON /AAAA2/ IWK, AMF(20), AOF(20), WK3(20), WK4(20), WK5(20)
   DIMENSION AMF1(1), AMF2(1), OMFR(1), EPS(7)

   NPH = 1
   NPH1 = 1
   NPH2 = 1
   SUM1 = 0.0
   SUM2 = 0.0
   DO 10, I = 1, NC

      SUM1 = SUM1 + AMF1(I)
      SUM2 = SUM2 + AMF2(I)

10 CONTINUE

   VAR = 1.0E-10
   IF (SUM1.GT.1.0+VAR) THEN

          CALL MFNM (NC,AMF1,AMF)
          DIFF = 0.0
          DO 20, I = 1, NC
             DIFF = DIFF + ABS(OMFR(I) - AMF(I))
20        CONTINUE
          DIFF = DIFF / FLOAT(NC)
          IF (DIFF.GT.EPS(7)) NPH1 = 2
```

```
            ENDIF

        IF (SUM2.GT.1.0+VAR) THEN

               CALL MFNM (NC,AMF2,AOF)
               DIFF = 0.0
               DO 30, I = 1, NC
                  DIFF = DIFF + ABS(OMFR(I) - AOF(I))
 30            CONTINUE
               DIFF = DIFF / FLOAT(NC)
               IF (DIFF.GT.EPS(7)) NPH2 = 2

            ENDIF

        IF ((NPH1.EQ.2).OR.(NPH2.EQ.2)) NPH = 2

        RETURN
        END
        SUBROUTINE PSPLT (NC,OMFR,AMF1,AMFR,AIV,WK,IER)
        COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
        DIMENSION OMFR(1), AMF1(1), AMFR(1), AIV(1), WK(1)

        IF (IPARM(26).NE.0) GO TO 10
        IPL = IPARM(5)
        IMET =  IPARM(4)

        CALL EQSOLI (NC,OMFR,AMF1,AMFR)
        IF (IPL.GE.1) CALL WRT3 (NC,AMF1,AMFR)
        IF (IMET.GE.2) GO TO 20

 10     CALL PSPACSS (NC,OMFR,AMF1,AMFR,AIV,ISW,IER)
        IF (ISW.EQ.1) GO TO 20
        GO TO 30

 20     CALL PSPHOM (NC,OMFR,AMF1,AMFR,AIV,WK,IER)

 30     RETURN
        END
        SUBROUTINE EQSOLI (NC,OMFR,AMF1,AMFR)
        COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
        COMMON /AAAA2/ IWK, AMN1(20), AMNR(20), WK3(20),
       1                    WK4(20), WK5(20)
        DIMENSION OMFR(1), AMF1(1), AMFR(1)

        SUM1 = 0.0
        SUM2 = 0.0
        DO 10, I = 1, NC

           SUM1 = SUM1 + AMF1(I)
           SUM2 = SUM2 + AMFR(I)
           AMN1(I) = AMF1(I)
```

```
      AMNR(I) = AMFR(I)

10 CONTINUE

   CALL MFNM (NC,AMN1,AMF1)
   CALL MFNM (NC,AMNR,AMFR)
   IF (SUM1.LT.1.0) CALL MFNM (NC,OMFR,AMF1)
   IF (SUM2.LT.1.0) CALL MFNM (NC,OMFR,AMFR)

   DIFF1 = 0.0
   DIFF2 = 0.0
   DO 20, I = 1, NC

      DIFF1 = DIFF1 + ABS(OMFR(I) - AMF1(I))
      DIFF2 = DIFF2 + ABS(OMFR(I) - AMFR(I))

20 CONTINUE

   DIFF1 = DIFF1 / FLOAT(NC)
   DIFF2 = DIFF2 / FLOAT(NC)

   IF ((DIFF1.GT.EPS(7)).AND.(DIFF2.GT.EPS(7))) GO TO 50
   IF (SUM1.GT.SUM2) THEN

          AMOLR = 1.0 / SUM1
          VAL = 1.0 - AMOLR
          IF (VAL.LT.1.0E-6) AMOLR = 0.49
          AMOL1 = 1.0 - AMOLR
          DO 30, I = 1, NC
             AMFR(I) = (OMFR(I) - AMOL1 * AMF1(I)) / AMOLR
30        CONTINUE

       ELSE

          AMOL1 = 1.0 / SUM2
          VAL = 1.0 - AMOL1
          IF (VAL.LT.1.0E-6) AMOL1 = 0.49
          AMOLR = 1.0 - AMOL1
          DO 40, I = 1, NC
             AMF1(I) = (OMFR(I) - AMOLR * AMFR(I)) / AMOL1
40        CONTINUE

       ENDIF

   GO TO 120

50 DIFF1 = 0.0
   DO 60, I = 1, NC
      DIFF1 = DIFF1 + ABS(AMF1(I) - AMFR(I))
60 CONTINUE
```

```
      IF (DIFF1.GT.EPS(7)) GO TO 80
      AMOL1 = 1.0 / SUM1
      VAL = 1.0 - AMOL1
      IF (VAL.LT.1.0E-6) AMOL1 = 0.49
      AMOLR = 1.0 - AMOL1
      DO 70, I = 1, NC
         AMFR(I) = (OMFR(I) - AMOL1 * AMF1(I)) / AMOLR
   70 CONTINUE
      GO TO 120

   80 AMOL1 =  (OMFR(1) - AMFR(1)) / (AMF1(1) - AMFR(1))
      IF ((AMOL1.LE.0.0).OR.(AMOL1.GE.1.0)) GO TO 90
      AMOLR = 1.0 - AMOL1
      GO TO 120

   90 IF (SUM1.GT.SUM2) THEN

             AMOLR = SUM2 / SUM1
             VAL = 1.0 - AMOLR
             IF (VAL.LT.1.0E-6) AMOLR = 0.49
             AMOL1 = 1.0 - AMOLR
             DO 100, I = 1, NC
                AMFR(I) = (OMFR(I) - AMOL1 * AMF1(I)) / AMOLR
  100        CONTINUE

          ELSE

             AMOL1 = SUM1 / SUM2
             VAL = 1.0 - AMOL1
             IF (VAL.LT.1.0E-6) AMOL1 = 0.49
             AMOLR = 1.0 - AMOL1
             DO 110, I = 1, NC
                AMF1(I) = (OMFR(I) - AMOLR * AMFR(I)) / AMOL1
  110        CONTINUE

          ENDIF

  120 IF (SUM1.GT.SUM2) THEN

             AMIN = 1.0
             DO 130, I = 1, NC

                VAL = OMFR(I) - AMOL1 * AMF1(I)
                IF (VAL.LE.AMIN) THEN

                       AMIN = VAL
                       IMIN = I

                    ENDIF

  130        CONTINUE
```

```
            IF (AMIN.LE.0.0) CALL CKSOL (NC,OMFR,AMF1,AMFR,
     1                                    IMIN,AMOL1,AMOLR)

         ELSE

            AMIN = 1.0
            DO 140, I = 1, NC

               VAL = OMFR(I) - AMOLR * AMFR(I)
               IF (VAL.LE.AMIN) THEN

                    AMIN = VAL
                    IMIN = I

                  ENDIF

 140        CONTINUE
            IF (AMIN.LE.0.0) CALL CKSOL (NC,OMFR,AMFR,AMF1,
     1                                    IMIN,AMOLR,AMOL1)

         ENDIF

      RETURN
      END
      SUBROUTINE PSPACSS (NC,OMFR,AMF1,AMFR,GN,ISW,IER)
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      COMMON /AAAA2/ IWK, EQR(20), EQRO(20), FGC1(20),
     1                    FGCR(20), WK5(20)
      COMMON /AAAA8/ OMN, SRO
      DIMENSION OMFR(1), AMF1(1), AMFR(1), GN(1)

      MAXIT = IPARM(9)
      IFC = IPARM(4)
      ISW = 0
      DO 10, I = 1, NC
         EQR(I) = AMFR(I) / AMF1(I)
   10 CONTINUE

      DO 30, I = 1, MAXIT

         NIT = I
         CALL PDIST2 (NC,OMFR,EQR,AMF1,AMFR)
         CALL FGCF (NC,AMF1,FGC1,WK5,Z)
         CALL FGCF (NC,AMFR,FGCR,WK5,Z)
         CALL ACCEL (NIT,NC,FGCR,AMFR,FGC1,AMF1,EQR,EQRO,GN,WK5,ACFTN)
         CALL CNVGPS (NC,NIT,EQR,EQRO,AMF1,FGC1,AMFR,FGCR,ICN)
         IF (ICN.EQ.1) GO TO 40
         IF (IFC.LT.0) GO TO 20
         GO TO 30
   20    CALL SWITCH (NC,NIT,AMF1,FGC1,AMFR,FGCR,ISW)
         IF (ISW.EQ.1) GO TO 40
```

```
 30 CONTINUE
    IER = 210
    GO TO 50

 40 IPARM(16) = NIT
    IER = 0

 50 RETURN
    END
    SUBROUTINE ACCEL (NIT,NC,FGCR,AMFR,FGC1,AMF1,EQR,EQRO,GN,
   1                  GNP1,ACFTN)
    DIMENSION FGCR(1), AMFR(1), FGC1(1), AMF1(1)
    DIMENSION EQR(1), EQRO(1), GN(1), GNP1(1)

    IF (NIT.GT.1) GO TO 20

    DO 10, I = 1, NC

       GN(I) = ALOG(FGCR(I) * AMFR(I) / (FGC1(I) * AMF1(I)))
       EQRO(I) = EQR(I)
       EQR(I) = FGC1(I) / FGCR(I)

 10 CONTINUE

    ACFTN = 1.0
    RETURN

 20 DO 30, I = 1, NC
       GNP1(I) = ALOG(FGCR(I) * AMFR(I) / (FGC1(I) * AMF1(I)))
 30 CONTINUE

    SUM1 = 0.0
    SUM2 = 0.0
    DO 40, I = 1, NC
       SUM1 = SUM1 + GN(I)**2
       SUM2 = SUM2 + GN(I) * GNP1(I)

 40 CONTINUE

    DN = SUM2 - SUM1
    IF (DN.EQ.0.0) DN = SUM1 * ACFTN
    ACFT = ABS(SUM1 * ACFTN / DN)

    DO 50, I = 1, NC

       CL = ABS(GNP1(I))
       FACT = ACFT * CL
       IF (FACT.GT.6.0) ACFT = 6.0 / CL

 50 CONTINUE
```

```
   IF (ACFT.LT.1.0) ACFT = 1.0

   DO 60, I = 1, NC

      EQRO(I) = EQR(I)
      EQR(I) = EQRO(I) * EXP(-ACFT * GNP1(I))
      GN(I) = GNP1(I)

60 CONTINUE

   ACFTN = ACFT

   RETURN
   END
   SUBROUTINE CNVGPS (NC,NIT,EQR,EQRO,AMF1,FGC1,AMFR,FGCR,ICN)
   COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
   DIMENSION EQR(1), EQRO(1), AMF1(1)
   DIMENSION FGC1(1), AMFR(1), FGCR(1)

   FNM = 0.0
   CHNG = 0.0
   DO 10, I = 1, NC

      CHNG = CHNG + (EQR(I) - EQRO(I))**2 / (EQR(I) * EQRO(I))
      VAL = ALOG(FGC1(I) * AMF1(I)) - ALOG(FGCR(I) * AMFR(I))
      FNM = FNM + VAL**2

10 CONTINUE

   IF ((CHNG.GT.EPS(3)).AND.(FNM.GT.EPS(3))) GO TO 20
   IF (NIT.LT.2) GO TO 20
   ICN = 1
   RPARM(3) = FNM
   RETURN

20 ICN = 0
   RETURN
   END
   SUBROUTINE SWITCH (NC,NIT,AMF1,FGC1,AMFR,FGCR,ISW)
   COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
   COMMON /AAAA8/ OMN, SRO
   DIMENSION AMF1(1), FGC1(1), AMFR(1), FGCR(1)

   ISW = 0
   IF (NIT.GT.1) GO TO 20

   OMN = AMOLR
   SRO = 0.0
   DO 10, I = 1, NC
      SRO = SRO + (FGC1(I) * AMF1(I) / (FGCR(I) * AMFR(I)) - 1.0)**2
```

```
 10 CONTINUE
    RETURN

 20 IC = 0
    IF (AMOLR.GE.AMOL1) THEN

           RTN = ABS(AMOLR - OMN) / OMN

        ELSE

           DN = 1.0 - OMN
           RTN = ABS(AMOL1 - DN) / DN
        ENDIF

    IF (RTN.LE.EPS(4)) IC = IC + 1
    SRN = 0.0
    DO 30, I = 1, NC
       SRN = SRN + (FGC1(I) * AMF1(I) / (FGCR(I) * AMFR(I)) - 1.0)**2
 30 CONTINUE

    IF  (SRN.LE.EPS(5)) IC = IC + 1
    RT = SRN / SRO
    IF (IPARM(4).GE.2) WRITE (7,100) NIT, RTN, SRN, RT
    IF (RT.LE.EPS(6)) IC = IC + 1
    IF (IC.EQ.3) GO TO 40
    SRO = SRN
    OMN = AMOLR
    RETURN

 40 ISW = 1
    RETURN

100 FORMAT (1X,3HNIT,I4,5X,3HRTN,E15.5,5X,3HSRN,E15.5,
   1   5X,2HRT,E15.5)

    END
    SUBROUTINE PSPHOM (NC,OMFR,AMF1,AMFR,AIV,WK,IER)
    COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
    COMMON /AAAA2/ NFE, WK1(20), WK2(20), WK3(20), WK4(20), WK5(20)
    DIMENSION OMFR(1), AMF1(1), AMFR(1)
    DIMENSION AIV(1), WK(1)
    EXTERNAL GFCN, FFCN

    MAXIT = IPARM(9) - IPARM(16)
    NSIG = IPARM(10)
    ISM = IPARM(11)
    NFE = 0
    IF (ISM.EQ.2) GO TO 40

    IF (AMOL1.LE.AMOLR) THEN
```

```
         DO 10, I = 1, NC
            AIV(I) = AMOL1 * AMF1(I)
10       CONTINUE

      ELSE

         DO 20, I = 1, NC
            AIV(I) = AMOLR * AMFR(I)
20       CONTINUE

      ENDIF

   IF (IABS(IPARM(4)).EQ.2) THEN

         CALL ZSCNT (GFCN,NSIG,NC,MAXIT,OMFR,AIV,FNM,WK,IER)
         IF (IER.NE.0) IER = 220

      ELSE

         CALL ZSPOW (GFCN,NSIG,NC,MAXIT,OMFR,AIV,FNM,WK,IER)
         IF (IER.NE.0) IER = 230

      ENDIF

   IF (IER.NE.0) GO TO 80

   CALL MFNM (NC,AIV,AMF1)
   AMOL1 = 0.0
   DO 30, I = 1, NC

      AMOL1 = AMOL1 + AIV(I)
      AIV(I) = OMFR(I) - AIV(I)

30 CONTINUE
   CALL MFNM (NC,AIV,AMFR)
   AMOLR = 1.0 - AMOL1

   GO TO 70

40 DO 50, I = 1, NC
      AIV(I) = ALOG(AMFR(I) / AMF1(I))
50 CONTINUE

   IF (IABS(IPARM(4)).EQ.2) THEN

         CALL ZSCNT (FFCN,NSIG,NC,MAXIT,OMFR,AIV,FNM,WK,IER)
         IF (IER.NE.0) IER = 220

      ELSE

         CALL ZSPOW (FFCN,NSIG,NC,MAXIT,OMFR,AIV,FNM,WK,IER)
```

```
         IF (IER.NE.0) IER = 230

      ENDIF

   IF (IER.NE.0) GO TO 80

   DO 60, I = 1, NC
      AIV(I) = EXP(AIV(I))
60 CONTINUE

   CALL PDIST2 (NC,OMFR,AIV,AMF1,AMFR)

70 IPARM(17) = NFE
   RPARM(3) = FNM

80 RETURN
   END
   SUBROUTINE GFCN (AIV,GON,NC,OMFR)
   COMMON /AAAA2/ NFE, AMF1(20), FGC1(20), AMF2(20),
  1                    FGC2(20), AMN(20)
   DIMENSION AIV(1), GON(1), OMFR(1)

   CALL CKMN (NC,AIV,OMFR)
   CALL MFNM (NC,AIV,AMF1)
   CALL FGCF (NC,AMF1,FGC1,AMN,Z)

   DO 10, I = 1, NC
      AMN(I) = OMFR(I) - AIV(I)
10 CONTINUE

   CALL MFNM (NC,AMN,AMF2)
   CALL FGCF (NC,AMF2,FGC2,AMN,Z)

   DO 20, I = 1, NC
      GON(I) = ALOG(AMF2(I) * FGC2(I)) - ALOG(AMF1(I) * FGC1(I))
20 CONTINUE

   NFE = NFE + 1
   RETURN
   END
   SUBROUTINE FFCN (AIV,FOK,NC,OMFR)
   COMMON /AAAA2/ NFE, AMF1(20), FGC1(20), AMF2(20),
  1                    FGC2(20), EQR(20)
   DIMENSION AIV(1), FOK(1), OMFR(1)

   DO 10, I = 1, NC
      EQR(I) = EXP(AIV(I))
10 CONTINUE

   CALL PDIST2 (NC,OMFR,EQR,AMF1,AMF2)
   CALL FGCF (NC,AMF1,FGC1,EQR,Z)
```

```
      CALL FGCF (NC,AMF2,FGC2,EQR,Z)

      DO 20, I = 1, NC
         FOK(I) = ALOG(AMF1(I) * FGC1(I)) - ALOG(AMF2(I) * FGC2(I))
   20 CONTINUE

      NFE  = NFE + 1
      RETURN
      END
      SUBROUTINE CKMN (NC,AMN,OMFR)
      DIMENSION AMN(1), OMFR(1)

      DO 10, I = 1, NC

         IF (AMN(I).LE.0.0) GO TO 20
         DIFF = OMFR(I) - AMN(I)
         IF (DIFF.LE.0.0) GO TO 20

   10 CONTINUE
      RETURN

   20 IER = 301
      CALL ERROPR (IER)

      RETURN
      END
      SUBROUTINE PDIST2 (NC,OMFR,EQR,AMF1,AMFR)
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      DIMENSION OMFR(1), EQR(1), AMF1(1), AMFR(1)

      MAXIT = IPARM(8)
      EPSPD = EPS(2)
      IF (AMOLR.GT.AMOL1) GO TO 60

      APV = AMOLR
      DO 20, I = 1, MAXIT

         NIT = I
         FOV = 0.0
         FPOV = 0.0
         DO 10, J = 1, NC

            CKM1 = EQR(J) - 1.0
            DN = 1.0 + APV * CKM1
            FOV = FOV + OMFR(J) * CKM1 / DN
            FPOV = FPOV - OMFR(J) * CKM1**2 / (DN * DN)

   10    CONTINUE
         IF (ABS(FOV).LE.EPSPD) GO TO 40
         IF (FPOV.EQ.0.0) GO TO 130
         DELV = FOV / FPOV
```

```
         IF (ABS(DELV).LE.EPSPD) GO TO 30
         APV = APV - DELV

 20   CONTINUE
      IER = 401
      GO TO 150

 30   APV = APV - DELV
 40   IF ((APV.LT.0.0).OR.(APV.GT.1.0)) GO TO 140
      AMOLR = APV
      AMOL1 = 1.0 - AMOLR
      DO 50, I = 1, NC

         AMF1(I) = OMFR(I) / (1.0 + AMOLR * (EQR(I) - 1.0))
         AMFR(I) = EQR(I) * AMF1(I)

 50   CONTINUE
      GO TO 120

 60   APV = AMOL1
      DO 80, I = 1, MAXIT

         NIT = I
         FOV = 0.0
         FPOV = 0.0
         DO 70, J = 1, NC

            CKM1 = 1.0 / EQR(J) - 1.0
            DN = 1.0 + APV * CKM1
            FOV = FOV - OMFR(J) * CKM1 / DN
            FPOV = FPOV + OMFR(J) * CKM1**2 / (DN * DN)

 70      CONTINUE
         IF (ABS(FOV).LE.EPSPD) GO TO 100
         IF (FPOV.EQ.0.0) GO TO 130
         DELV = FOV / FPOV
         IF (ABS(DELV).LE.EPSPD) GO TO 90
         APV = APV - DELV

 80   CONTINUE
      IER = 401
      GO TO 150

 90   APV = APV - DELV
100   IF ((APV.LT.0.0).OR.(APV.GT.1.0)) GO TO 140
      AMOL1 = APV
      AMOLR = 1.0 - AMOL1
      DO 110, I = 1, NC

         AMFR(I) = OMFR(I) / (1.0 + AMOL1 * (1.0 / EQR(I) - 1.0))
         AMF1(I) = AMFR(I) / EQR(I)
```

```
110 CONTINUE

120 IPARM(18) = IPARM(18) + NIT
    RPARM(4) = FOV
    RETURN

130 IER = 402
    GO TO 150

140 IER = 403
150 CALL ERROPR (IER)

    RETURN
    END
    SUBROUTINE WRT1 (NC,EQR,AMF1,AMF2)
    DIMENSION EQR(1), AMF1(1), AMF2(1)

    WRITE (7,100)
    WRITE (7,110) (I, EQR(I), AMF1(I), AMF2(I), I = 1, NC)

    RETURN

100 FORMAT (//,1X,*STARTING INFORMATION FOR PHASE STABILITY ANALYSIS*,
   1   //,2X,9HCOMPONENT,9X,11HEQUILIBRIUM,2(7X,13HMOLE FRACTION),
   2   /,6X,5HINDEX,15X,5HRATIO,10X,10HSOLUTION 1,10X,
   3   10HSOLUTION 2,/,1X,10H----------,3(5X,15H---------------),/)
110 FORMAT (1X,I10,3E20.7)

    END
    SUBROUTINE WRT2 (NC,OMFR,AMF1,AMF2)
    DIMENSION OMFR(1), AMF1(1), AMF2(1)

    WRITE (7,100)

    SUM1 = 0.0
    SUM2 = 0.0
    DO 10, I = 1, NC

       WRITE(7,110) I, OMFR(I), AMF1(I), AMF2(I)
       SUM1 = SUM1 + AMF1(I)
       SUM2 = SUM2 + AMF2(I)

 10 CONTINUE

    WRITE (7,120) SUM1, SUM2

    RETURN

100 FORMAT (//,1X,*PHASE STABILITY ANALYSIS RESULTS:*,
   1   //,2X,9HCOMPONENT,14X,6HGLOBAL,10X,10HSOLUTION 1,
```

```
   2   10X,10HSOLUTION 2,/,6X,5HINDEX,3(9X,11HCOMPOSITION ),
   3   /,1X,10H---------- ,3(5X,15H--------------- ),/)
110 FORMAT (1X,I10,3E20.12)
120 FORMAT (/,1X,*MOLE FRACTION SUM*,13X,2E20.12,
   1   /,1X,*(AFTER CONVERGENCE)*)

    END
    SUBROUTINE WRT3 (NC,AMF1,AMFR)
    COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
    DIMENSION AMF1(1), AMFR(1)

    WRITE (7,100)
    WRITE (7,110) AMOL1, AMOLR
    WRITE (7,120)
    WRITE (7,130) (I, AMF1(I), AMFR(I), I = 1, NC)

    RETURN

100 FORMAT (//,1X,*STARTING INFORMATION FOR PHASE SPLIT CALCULATION*)
110 FORMAT (/,1X,*PHASE MOLES, 1*,5X,E20.10,
   1   /,1X,*PHASE MOLES, R*,5X,E20.10)
120 FORMAT (/,2X,9HCOMPONENT,13X,7HPHASE 1,13X,7HPHASE R,
   1   /,6X,5H1NDEX,2(9X,11HCOMPOSITION),/,1X,10H---------- ,
   2   2(5X,15H--------------- ),/)
130 FORMAT (1X,I10,2E20.12)

    END
    SUBROUTINE FGCF (N,AMFR,FUGC,WK,Z)
    COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
    COMMON /AAAA3/ TEM, PRS
    COMMON /AAAA5/ AI(20), BI(20), AIJ(20,20)
    DIMENSION AMFR(1), FUGC(1), WK(1)

    IEOS = IPARM(1)
    A = 0.0
    B = 0.0
    DO 20, I = 1, N

       B = B + AMFR(I) * BI(I)
       DO 10, J = 1, N
          A = A + AMFR(I) * AMFR(J) * AIJ(I,J)

 10    CONTINUE

 20 CONTINUE

    R = 10.732
    BB = B * PRS / (TEM * R)
    AA = A * PRS / (TEM * R)**2

    IF (IEOS.EQ.1) THEN
```

```
            A2 = BB - 1.0
            A1 = AA - BB * (3.0 * BB + 2.0)
            A0 = BB * (BB * (BB + 1.0) - AA)

         ELSE

            A2 = -1.0
            A1 = AA - BB * (1.0 + BB)
            A0 = -AA * BB

         ENDIF

      CALL ZCALC (A2,A1,A0,Z1,Z2,Z3,IJZ)

      IF (IJZ.LT.0) GO TO 90
      IF (IJZ.EQ.3) GO TO 30
      IF (Z1.LE.0.0) GO TO 70
      IF (Z2.LE.0.0) GO TO 70
      ZL = AMIN1(Z1,Z2)
      ZH = AMAX1(Z1,Z2)
      GO TO 40

   30 ZH = AMAX1 (Z1,Z2,Z3)
      ZL = ZH
      IF ((Z1.GT.0.0).AND.(Z1.LE.ZL)) ZL = Z1
      IF ((Z2.GT.0.0).AND.(Z2.LE.ZL)) ZL = Z2
      IF ((Z3.GT.0.0).AND.(Z3.LE.ZL)) ZL = Z3

   40 CALL SUBPHI (AA,BB,N,IEOS,ZH,AMFR,FUGC,A,B)
      CALL SUBPHI (AA,BB,N,IEOS,ZL,AMFR,WK,A,B)

      SUM = 0.0
      DO 50, I = 1, N
         SUM = SUM + AMFR(I) * (FUGC(I) - WK(I))
   50 CONTINUE

      IF (SUM.LT.0.0) THEN

            Z = ZH
            DO 61, I = 1, N
               FUGC(I) = EXP(FUGC(I))
   61       CONTINUE

         ELSE

            Z = ZL
            DO 62, I = 1, N
               FUGC(I) = EXP(WK(I))
   62       CONTINUE
```

```
          ENDIF

      RETURN

   70 Z = AMAX1(Z1,Z2)
      CALL SUBPHI (AA,BB,N,IEOS,Z,AMFR,FUGC,A,B)
      DO 80, I = 1, N
         FUGC(I) = EXP(FUGC(I))
   80 CONTINUE
      RETURN

   90 CALL SUBPHI (AA,BB,N,IEOS,Z1,AMFR,FUGC,A,B)
      Z = Z1
      DO 100, I = 1, N
         FUGC(I) = EXP(FUGC(I))
  100 CONTINUE
      RETURN

      END
      SUBROUTINE ZCALC (A2,A1,A0,Z1,Z2,Z3,IJZ)

C     THIS SUBROUTINE CALCULATES Z1, Z2 AND Z3 WHICH ARE ROOTS TO
C    THE EQUATION OF STATE WHEN WRITTEN IN TERMS OF Z.  THE EQUATION
C    RESULTS IN A CUBIC IN Z.  IN THIS PROGRAM, THE VALUES OF THE COEF-
C    FICIENTS ARE PASSED TO THIS SUBROUTINE.  HERE Z1, Z2 AND Z3 ARE
C    COMPUTED ANALYTICALLY BY USING CARDANO'S METHOD, IN WHICH THE
C    ORIGINAL CUBIC  IS REDUCED TO THE FORM:
C
C              Y**3 + PY + Q = 0,      Z = Y - A2 / 3
C
C    THERE ARE THREE POSSIBLE SITUATIONS FOR THE THREE ROOTS.  THEY ARE
C    EXPLAINED BELOW.

      P3 = A1 / 3.0 - A2 * A2 / 9.0
      Y = A2 / 3.0
      Q2 = Y**3 - A1 * A2 / 6.0 + A0 / 2.0
      Q = Q2 * Q2 + P3**3

C     THE SCENARIO OF THE ROOTS DEPENDS ON Q.

      IF (Q.GT.0.0) THEN

             U = (ABS(Q**0.5 - Q2))**(1.0 / 3.0)
             IF (Q**0.5.LT.Q2) U = - U
             Z1 = U - (P3 / U + Y)
             Z2 = 0.0
             Z3 = 0.0
             IJZ = -1
C     HERE, IF Q>0, THERE IS ONE REAL ROOT AND A PAIR OF COMPLEX
C    ROOTS WHICH ARE ESSENTIALLY DISCARDED BY BEING SET = 0.
```

```
          ELSE
             IF (ABS(Q).LT.1.0E-11) THEN

                   Z1 = 2.0 * Q2 / P3 - Y
                   Z2 = Q2 * P3 -Y
                   Z3 = 0.0
                   IJZ = 2
C       IN THIS CASE, Q=0, AND THERE ARE THREE REAL ROOTS, TWO
C      OF WHICH ARE EQUAL.  THE SECOND IS DISCARDED.

                 ELSE

C       IT IS NO LONGER EXPEDIENT TO CALCULATE THE LAST CASE IN
C      WHICH THERE ARE THREE DIFFERENT, REAL ROOTS BY THE ABOVE
C      PROCEDURES.  INSTEAD A TRIGONOMETRIC SOLUTION IS USED.

                    PI3 = ACOS(-1.0) / 3.0
                    ARAD = SQRT(- (P3**3))
                    ALPHCS = - Q2 / ARAD
                    ALPHA = ACOS(ALPHCS)
                    RADL = SQRT(- P3)
                    Z1 = 2.0 * RADL * COS(ALPHA / 3.0) - Y
                    Z2 = -2.0*RADL*COS(ALPHA/3.0 + PI3) - Y
                    Z3 = -2.0*RADL*COS(ALPHA/3.0 - PI3) - Y
                    IJZ = 3

                 ENDIF
              ENDIF

        RETURN
        END
        SUBROUTINE SUBPHI (AA,BB,N,IEOS,Z,AMFI,ALNPHI,A,B)
        COMMON /AAAA5/ AI(20), BI(20), AIJ(20,20)
        DIMENSION AMFI(1), ALNPHI(1)

        AH = ALOG (Z - BB)
        IF (IEOS.EQ.1) THEN

              AJ = AA / (2.0 * BB * SQRT(2.0))
              C1 = 1.0 + SQRT(2.0)
              C2 = 1.0 - SQRT(2.0)
              AK = (Z + C1 * BB) / (Z + C2 * BB)

           ELSE

              AJ = AA / BB
              AK = (Z + BB) / Z

           ENDIF

        DO 60, I = 1, N
```

```
      SM = 0.0
      DO 50, J = 1, N
         SM = SM + AMFI(J) * AIJ(J,I)
50    CONTINUE

      SUMAIK = 2.0 * SM
      AL = AJ * (SUMAIK / A - BI(I) / B) * ALOG (AK)
      ALNPHI(I) = BI(I) * (Z - 1.0) / B - (AH + AL)
60 CONTINUE

   RETURN
   END
   SUBROUTINE CKSOL (NC,OMFR,AMFI,AMFD,IMIN,AMOLI,AMOLD)
   DIMENSION OMFR(1), AMFI(1), AMFD(1)

   AMOLI = 0.99 * OMFR(IMIN) / AMFI(IMIN)
   AMOLD = 1.0 - AMOLI
   DO 10, I = 1, NC
      AMFD(I) = (OMFR(I) - AMOLI * AMFI(I)) / AMOLD
10 CONTINUE

   RETURN
   END
   SUBROUTINE PSATP (NC,AMF1,AMF2,AMFR,HG,WK,NPH,IER)
   COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
   COMMON /AAAA7/ SOL1(20), SOL2(20), WK1(20)
   DIMENSION AMF1(1), AMF2(1), AMFR(1), HG(1), WK(1)

   IS = 12
   IF (IPARM(3).GT.1) IS = 14
   IT1 = IPARM(IS)
   IT2 = IPARM(IS+1)
   IPL = IPARM(5)
   IMET = IPARM(3)

   CALL SETH (NC,AMF1,HG,WK1)
   CALL SETV1 (NC,SOL1,SOL2)
   IF (IMET.GE.2) GO TO 10
   CALL PSASS (NC,HG,SOL1,SOL2,IER)
   GO TO 20

10 CALL PSAHOM (NC,AMF1,HG,SOL1,SOL2,WK,WK1,IER)
20 IF (IER.NE.0) GO TO 90

   CALL SOLAN2 (NC,AMF1,AMFR,EPS,SOL1,SOL2,WK1,ITP,ISOLI)
   IPARM(19) = IPARM(IS)
   IPARM(20) = IPARM(IS+1)
   IF (IPL.GE.1) CALL WRT2 (NC,AMF1,SOL1,SOL2)
   IF (ITP.EQ.1) GO TO 50

   CALL SETV2 (NC,AMF1,AMFR,HG,SOL1,SOL2,WK1)
```

```
   IF (IMET.GE.2) GO TO 30
   CALL PSASS (NC,HG,SOL1,SOL2,IER)
   GO TO 40

30 CALL PSAHOM (NC,AMF1,HG,SOL1,SOL2,WK,WK1,IER)
40 IF (IER.NE.0) GO TO 90

   CALL SOLAN2 (NC,AMF1,AMFR,EPS,SOL1,SOL2,WK1,ITP,ISOLI)
   IPARM(21) = IPARM(IS)
   IPARM(22) = IPARM(IS+1)
   IPARM(IS) = IT1
   IPARM(IS+1) = IT2
   IF (IPL.GE.1) CALL WRT2 (NC,AMF1,SOL1,SOL2)
   IF (ITP.EQ.0) GO TO 80

50 IF (ISOLI.EQ.1) THEN

          DO 60, I = 1, NC
             AMF2(I) = SOL1(I)
60        CONTINUE

       ELSE

          DO 70, I = 1, NC
             AMF2(I) = SOL2(I)
70        CONTINUE

       ENDIF

   NPH = 3
   RETURN

80 NPH = 2
90 RETURN
   END
   SUBROUTINE SETV1 (NC,SOL1,SOL2)
   DIMENSION SOL1(1), SOL2(1)

   FR = (1.0 - .999) / FLOAT(NC-1)
   DO 10, I = 1, NC
      SOL1(I) = FR
      SOL2(I) = FR
10 CONTINUE

   SOL1(1) = 1.0 - FR * FLOAT(NC-1)
   SOL2(NC) = 1.0 - FR * FLOAT(NC-1)

   RETURN
   END
   SUBROUTINE SETV2 (NC,AMF1,AMFR,HG,SOL1,SOL2,WK1)
   DIMENSION AMF1(1), AMFR(1), HG(1), SOL1(1), SOL2(1), WK1(1)
```

```
      DO 10, I = 1, NC
         WK1(I) = EXP(HG(I))
   10 CONTINUE

      CALL MFNM (NC,WK1,SOL1)

      DO 20, I = 1, NC
         WK1(I) = 0.5 * (AMF1(I) + AMFR(I))
   20 CONTINUE

      CALL MFNM (NC,WK1,SOL2)

      RETURN
      END
      SUBROUTINE SOLAN2 (NC,AMF1,AMFR,EPS,SOL1,SOL2,AMF,ITP,ISI)
      DIMENSION AMF1(1), AMFR(1), SOL1(1)
      DIMENSION SOL2(1), EPS(8), AMF(1)

      ITP = 0
      SUM1 = 0.0
      SUM2 = 0.0
      DO 10, I = 1, NC

         SUM1 = SUM1 + SOL1(I)
         SUM2 = SUM2 + SOL2(I)

   10 CONTINUE

      IF (SUM1.GT.1.0) THEN

             CALL MFNM (NC,SOL1,AMF)
             DIFF1 = 0.0
             DIFF2 = 0.0
             DO 20, I = 1, NC

                DIFF1 = DIFF1 + ABS(AMF1(I) - AMF(I))
                DIFF2 = DIFF2 + ABS(AMFR(I) - AMF(I))

   20        CONTINUE
             DIFF1 = DIFF1 / FLOAT(NC)
             DIFF2 = DIFF2 / FLOAT(NC)
             IF ((DIFF1.GT.EPS(7)).AND.(DIFF2.GT.EPS(8))) ITP = 1

         ENDIF

      IF (ITP.NE.0) GO TO 40

      IF (SUM2.GT.1.0) THEN

             CALL MFNM (NC,SOL2,AMF)
```

```
               DIFF1 = 0.0
               DIFF2 = 0.0
               DO 30, I = 1, NC

                  DIFF1 = DIFF1 + ABS(AMF1(I) - AMF(I))
                  DIFF2 = DIFF2 + ABS(AMFR(I) - AMF(I))

30             CONTINUE
               DIFF1 = DIFF1 / FLOAT(NC)
               DIFF2 = DIFF2 / FLOAT(NC)
               IF ((DIFF1.GT.EPS(7)).AND.(DIFF2.GT.EPS(8))) ITP = 1

            ENDIF

      IF (ITP.NE.0) GO TO 50
      RETURN

40    ISI = 1
      RETURN

50    ISI = 2
      RETURN
      END
      SUBROUTINE PSPLT3 (NC,OMFR,AMF1,AMF2,AMFR,WRK1,WRK2,IER)
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      DIMENSION OMFR(1), AMF1(1), AMF2(1), AMFR(1)
      DIMENSION WRK1(1), WRK2(1)

      IPL = IPARM(5)
      IMET = IPARM(4)

      CALL EQSOL3 (NC,OMFR,AMF1,AMF2,AMFR,WRK1,WRK2)
      IF (IPL.GE.1) CALL WRT4 (NC,AMF1,AMF2,AMFR)
      IF (IMET.GE.2) GO TO 20
      CALL PSPASS3 (NC,OMFR,AMF1,AMF2,AMFR,WRK1,WRK2,ISW,IER)
      IF (ISW.EQ.1) GO TO 10
      GO TO 20

10    CALL PSPHOM3 (NC,OMFR,AMF1,AMF2,AMFR,IER)
20    RETURN
      END
      SUBROUTINE PSPASS3 (NC,OMFR,AMF1,AMF2,AMFR,GN1,GN2,ISW,IER)
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      COMMON /AAAA2/ IWK, EQR1(20), EQR2(20), FGC1(20), FGC2(20),
     1               FGCR(20)
      COMMON /AAAA7/ EQR1N(20), EQR2N(20), WRK(20)
      COMMON /AAA10/ IPCFC, RESN, PFGC(20)
      DIMENSION OMFR(1), AMF1(1), AMF2(1), AMFR(1)
      DIMENSION GN1(1), GN2(1)

      MAXIT = IPARM(9)
```

```
      IFC = IPARM(4)
      ISW = 0

      DO 10, I = 1, NC

         EQR1(I) = AMFR(I) / AMF1(I)
         EQR2(I) = AMFR(I) / AMF2(I)

   10 CONTINUE

      DO 30, I = 1, MAXIT

         NIT = I
         CALL PDIST3 (NC,OMFR,EQR1,EQR2)
         CALL MSSCN (NC,OMFR,EQR1,EQR2,AMF1,AMF2,AMFR)
         CALL FGCF (NC,AMF1,FGC1,WRK,Z)
         CALL FGCF (NC,AMF2,FGC2,WRK,Z)
         CALL FGCF (NC,AMFR,FGCR,WRK,Z)
         CALL ACCEL3 (NC,NIT,AMF1,AMF2,AMFR,FGC1,FGC2,FGCR,EQR1,EQR2,
     1         EQR1N,EQR2N,GN1,GN2,ACFTN)
         CALL CNVG3 (NC,NIT,EQR1,EQR2,AMF1,AMF2,AMFR,FGC1,FGC2,FGCR,
     1         EQR1N,EQR2N,ICN)
         IF (ICN.EQ.1) GO TO 40
         IF (ICN.LT.0) GO TO 50
         IF (IFC.LT.0) GO TO 20
         GO TO 30
   20    CALL SWITCH3 (NC,NIT,AMF1,AMF2,AMFR,FGC1,FGC2,FGCR,ISW)
         IF (ISW.EQ.1) GO TO 40

   30 CONTINUE
      IER = 210
      GO TO 60

   40 IPARM(23) = NIT
      IER = 0
      GO TO 60

   50 IPARM(23) = NIT
      IPARM(26) = 1

   60 RETURN
      END
      SUBROUTINE EQSOL3 (NC,OMFR,AMF1,AMF2,AMFR,AMN,SUM)
      COMMON /AAAA1/ IPARM(26), RPARM(6), EPS(8), AMOL1, AMOL2, AMOLR
      DIMENSION OMFR(1), AMF1(1), AMF2(1), AMFR(1)
      DIMENSION AMN(1), SUM(1)

      DO 10, I = 1, NC
         AMN(I) = AMF2(I)
   10 CONTINUE
```

```
      CALL MFNM (NC,AMN,AMF2)

      DO 20, I = 1, NC
         SUM(I) = AMF1(I) + AMF2(I) + AMFR(I)
   20 CONTINUE

      AM1 = 0.0
      AM2 = 0.0
      DO 30, I = 1, NC

         SUMMF = SUM(I)
         AMN(I) = AMF1(I) * OMFR(I) / SUMMF
         SUM(I) = AMF2(I) * OMFR(I) / SUMMF
         AM1 = AM1 + AMN(I)
         AM2 = AM2 + SUM(I)

   30 CONTINUE

      CALL MFNM (NC,AMN,AMF1)
      CALL MFNM (NC,SUM,AMF2)

      AMR = 1.0 - (AM1 + AM2)
      DO 40, I = 1, NC
         AMFR(I) = (OMFR(I) - AM1 * AMF1(I) - AM2 * AMF2(I)) / AMR
   40 CONTINUE

      AMOL1 = AM1
      AMOL2 = AM2
      AMOLR = AMR

      RETURN
      END
      SUBROUTINE SWITCH3 (NC,NIT,AMF1,AMF2,AMFR,FGC1,FGC2,FGCR,ISW)
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      COMMON /AAAA9/ SRO, OMN1, OMN2, OMNR
      DIMENSION AMF1(1), AMF2(1), AMFR(1)
      DIMENSION FGC1(1), FGC2(1), FGCR(1)

      ISW = 0
      IF (NIT.GT.1) GO TO 20

      OMN1 = AMOL1
      OMN2 = AMOL2
      OMNR = AMOLR
      SRO = 0.0
      DO 10, I = 1, NC

         VAL1 = (FGC1(I) * AMF1(I) / (FGCR(I) * AMFR(I)) - 1.0)**2
         VAL2 = (FGC2(I) * AMF2(I) / (FGCR(I) * AMFR(I)) - 1.0)**2
         SRO = SRO + VAL1 + VAL2
```

```
10 CONTINUE
   RETURN

20 IC = 0
   RT1 = ABS(AMOL1 - OMN1) / OMN1
   RT2 = ABS(AMOL2 - OMN2) / OMN2
   RTR = ABS(AMOLR - OMNR) / OMNR
   RT = AMAX1 (RT1,RT2)
   RTMAX = AMAX1 (RT,RTR)

   IF (RTMAX.LE.EPS(4)) IC = IC + 1

   SRN = 0.0
   DO 30, I = 1, NC

      VAL1 = (FGC1(I) * AMF1(I) / (FGCR(I) * AMFR(I)) - 1.0)**2
      VAL2 = (FGC2(I) * AMF2(I) / (FGCR(I) * AMFR(I)) - 1.0)**2
      SRN = SRN + VAL1 + VAL2

30 CONTINUE

   IF (SRN.LE.EPS(5)) IC = IC + 1
   RT = SRN / SRO
   IF (RT.LE.EPS(6)) IC = IC + 1
   IF (IC.EQ.3) GO TO 40

   SRO = SRN
   OMN1 = AMOL1
   OMN2 = AMOL2
   OMNR = AMOLR
   RETURN

40 ISW = 1
   RETURN
   END
   SUBROUTINE PDIST3 (NC,OMFR,EQR1,EQR2)
   COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
   DIMENSION OMFR(1), EQR1(1), EQR2(1)

   MAXIT = IPARM(8)
   EPSPD = EPS(2)
   AM1 = AMOL1
   AM2 = AMOL2
   DO 20, I = 1, MAXIT

      NIT = I
      G1 = 0.0
      G2 = 0.0
      PG1N1 = 0.0
      PG2N2 = 0.0
      PG1N2 = 0.0
```

```
      DO 10, J = 1, NC

         C1 = 1.0 / EQR1(J) - 1.0
         C2 = 1.0 / EQR2(J) - 1.0
         DN = 1.0 + AM1 * C1 + AM2 * C2
         G1 = G1 + C1 * OMFR(J) / DN
         G2 = G2 + C2 * OMFR(J) / DN
         PG1N1 = PG1N1 - C1**2 * OMFR(J) / DN**2
         PG2N2 = PG2N2 - C2**2 * OMFR(J) / DN**2
         PG1N2 = PG1N2 - C1 * C2 * OMFR(J) / DN**2

10    CONTINUE

      A = G1**2
      B = G2**2
      ER = SQRT(A+B)
      IF (ER.LE.EPSPD) GO TO 40
      DN = PG1N1 * PG2N2 - PG1N2**2
      IF (DN.EQ.0.0) GO TO 50
      A = (G1 * PG2N2 - G2 * PG1N2) / DN
      B = (G2 * PG1N1 - G1 * PG1N2) / DN
      ER = SQRT(A * A + B * B)
      IF (ER.LE.EPSPD) GO TO 30
      AM1 = AM1 - A
      AM2 = AM2 - B

20 CONTINUE
   IER = 401
   GO TO 70

30 AM1 = AM1 - A
   AM2 = AM2 - B
40 IF ((AM1.LT.0.0).OR.(AM1.GT.1.0)) GO TO 60
   IF ((AM2.LT.0.0).OR.(AM2.GT.1.0)) GO TO 60
   IF (AM1+AM2.GT.1.0) GO TO 60

   AMOL1 = AM1
   AMOL2 = AM2
   AMOLR = 1.0 - (AMOL1 + AMOL2)
   IPARM(25) = IPARM(25) + NIT
   RPARM(5) = ER
   RETURN

50 IER = 402
   GO TO 70

60 IER = 403
70 CALL ERROPR (IER)

   RETURN
   END
```

```
   SUBROUTINE MSSCN (NC,OMFR,EQR1,EQR2,AMF1,AMF2,AMFR)
   COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
   DIMENSION OMFR(1), EQR1(1), EQR2(1)
   DIMENSION AMF1(1), AMF2(1), AMFR(1)

   DO 10, I = 1, NC

      SUM = AMOL1 / EQR1(I) + AMOL2 / EQR2(I) + AMOLR
      F1 = 1.0 / (EQR1(I) * SUM)
      F2 = 1.0 / (EQR2(I) * SUM)
      FR = 1.0 / SUM
      AMF1(I) = F1 * OMFR(I)
      AMF2(I) = F2 * OMFR(I)
      AMFR(I) = FR * OMFR(I)

10 CONTINUE

   SUM1 = 0.0
   SUM2 = 0.0
   SUMR = 0.0
   DO 20, I = 1, NC

      SUM1 = SUM1 + AMF1(I)
      SUM2 = SUM2 + AMF2(I)
      SUMR = SUMR + AMFR(I)

20 CONTINUE

   DO 30, I = 1, NC

      AMF1(I) = AMF1(I) / SUM1
      AMF2(I) = AMF2(I) / SUM2
      AMFR(I) = AMFR(I) / SUMR

30 CONTINUE

   RETURN
   END
   SUBROUTINE ACCEL3 (NC,NIT,AMF1,AMF2,AMFR,FGC1,FGC2,FGCR,EQR1,EQR2,
  1                   EQR1N,EQR2N,GN1,GN2,ACFTN)
   DIMENSION AMF1(1), AMF2(1), AMFR(1), FGC1(1), FGC2(1), FGCR(1)
   DIMENSION EQR1(1), EQR2(1), EQR1N(1), EQR2N(1), GN1(1), GN2(1)

   IF (NIT.GT.1) GO TO 20

   DO 10, I = 1, NC

      GN1(I) = ALOG(FGCR(I) * AMFR(I)) - ALOG(FGC1(I) * AMF1(I))
      EQR1N(I) = EQR1(I)
      EQR1(I) = FGC1(I) / FGCR(I)
      GN2(I) = ALOG(FGCR(I) * AMFR(I)) - ALOG(FGC2(I) * AMF2(I))
```

```
      EQR2N(I) = EQR2(I)
      EQR2(I) = FGC2(I) / FGCR(I)

10 CONTINUE

   ACFTN = 1.0
   RETURN

20 DO 30, I = 1, NC

      EQR1N(I) = EQR1(I)
      EQR1(I) = ALOG(FGCR(I) * AMFR(I)) - ALOG(FGC1(I) * AMF1(I))
      EQR2N(I) = EQR2(I)
      EQR2(I) = ALOG(FGCR(I) * AMFR(I)) - ALOG(FGC2(I) * AMF2(I))

30 CONTINUE

   SUM1 = 0.0
   SUM2 = 0.0
   DO 40, I = 1, NC

      SUM1 = SUM1 + GN1(I)**2 + GN2(I)**2
      SUM2 = SUM2 + GN1(I) * EQR1(I) + GN2(I) * EQR2(I)

40 CONTINUE

   DN = SUM2 - SUM1
   IF (DN.EQ.0.0) DN = SUM1 * ACFTN
   ACFT = ABS(SUM1 * ACFTN / DN)

   DO 50, I = 1, NC

      C1 = ABS(EQR1(I))
      C2 = ABS(EQR2(I))
      CL = AMAX1 (C1,C2)
      FACT = ACFT * CL
      IF (FACT.GT.6.0) ACFT = 6.0 / CL

50 CONTINUE

   IF (ACFT.LT.1.0) ACFT = 1.0

   DO 60, I = 1, NC

      GNP = EQR1(I)
      EQR1(I) = EQR1N(I) * EXP(-ACFT * GNP)
      GN1(I) = GNP
      GNP = EQR2(I)
      EQR2(I) = EQR2N(I) * EXP(-ACFT * GNP)
      GN2(I) = GNP
```

```
60 CONTINUE

   ACFTN = ACFT

   RETURN
   END
   SUBROUTINE CNVG3 (NC,NIT,EQR1,EQR2,AMF1,AMF2,AMFR,FGC1,FGC2,FGCR,
  1                  EQR1N,EQR2N,ICN)
   COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
   COMMON /AAA10/ IPCFC, RESN, PFGC(20)
   DIMENSION EQR1(1), EQR2(1), AMF1(1), AMF2(1), AMFR(1)
   DIMENSION FGC1(1), FGC2(1), FGCR(1), EQR1N(1), EQR2N(1)

   FNM = 0.0
   CHNG = 0.0
   DO 10, I = 1, NC

      VAL1 = (EQR1(I) - EQR1N(I))**2 / (EQR1(I) * EQR1N(I))
      VAL2 = (EQR2(I) - EQR2N(I))**2 / (EQR2(I) * EQR2N(I))
      CHNG = CHNG + VAL1 + VAL2
      VAL1 = ALOG(FGC1(I) * AMF1(I)) - ALOG(FGCR(I) * AMFR(I))
      VAL2 = ALOG(FGC2(I) * AMF2(I)) - ALOG(FGCR(I) * AMFR(I))
      FNM = FNM + VAL1**2 + VAL2**2

10 CONTINUE

   IF ((CHNG.GT.EPS(3)).AND.(FNM.GT.EPS(3))) GO TO 20
   ICN = 1
   RPARM(6) = FNM
   RETURN

20 IF (NIT.GT.1) GO TO 30
   RESN = SQRT(FNM)
   ICN = 0
   IPCFC = 0
   RETURN

30 RESNP1 = SQRT(FNM)
   IF (RESNP1.GT.RESN) GO TO 40
   RESN = RESNP1
   ICN = 0
   RETURN

40 CALL EXPHCK (NC,AMF1,AMF2,AMFR,FGC1,FGC2,FGCR,IEP)
   IF (IEP.NE.0) GO TO 50
   ICN = 0
   RESN = RESNP1
   RETURN

50 ICN = -1
   RETURN
```

```
      END
      SUBROUTINE PSPHOM3 (NC,OMFR,AMF1,AMF2,AMFR,IER)
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      COMMON /AAAA2/ NFE, AM1(20), AM2(20), AMR(20), WRK4(20), WRK5(20)
      DIMENSION OMFR(1), AMF1(1), AMF2(1), AMFR(1)
      DIMENSION AIV(40), WK(5248)
      EXTERNAL GFCN3, FFCN3

      NC2 = 2 * NC
      MAXIT = IPARM(9) - IPARM(16)
      NSIG = IPARM(10)
      ISM = IPARM(11)
      IF (ISM.EQ.2) GO TO 60

      IF (AMOL1.LE.AMOLR) THEN

             IF (AMOL2.LE.AMOLR) THEN

                    DO 10, I = 1, NC

                       AIV(I) = AMOL1 * AMF1(I)
                       AIV(NC+I) = AMOL2 * AMF2(I)

10                  CONTINUE

                 ELSE

                    DO 20, I = 1, NC

                       AIV(I) = AMOL1 * AMF1(I)
                       AIV(NC+I) = AMOLR * AMFR(I)

20                  CONTINUE

                 ENDIF

          ELSE

             IF (AMOL1.LE.AMOL2) THEN

                    DO 30, I = 1, NC

                       AIV(I) = AMOLR * AMFR(I)
                       AIV(NC+I) = AMOL1 * AMF1(I)

30                  CONTINUE

                  ELSE

                     DO 40, I = 1, NC
```

```
                     AIV(I) = AMOLR * AMFR(I)
                     AIV(NC+I) = AMOL2 * AMF2(I)

40                 CONTINUE

               ENDIF


           ENDIF

   IF (IABS(IPARM(4)).EQ.2) THEN

          CALL ZSCNT (GFCN3,NSIG,NC2,MAXIT,OMFR,AIV,FNM,WK,IER)
          IF (IER.NE.0) IER = 220

       ELSE

          CALL ZSPOW (GFCN3,NSIG,NC2,MAXIT,OMFR,AIV,FNM,WK,IER)
          IF (IER.NE.0) IER = 230

       ENDIF
   IF (IER.NE.0) GO TO 100

   AMOL1 = 0.0
   AMOL2 = 0.0
   DO 50, I = 1, NC

      AM1(I) = AIV(I)
      AM2(I) = AIV(NC+I)
      AMOL1 = AMOL1 + AM1(I)
      AMOL2 = AMOL2 + AM2(I)
      AMR(I) = AM1(I) + AM2(I)

50 CONTINUE

   CALL MFNM (NC,AM1,AMF1)
   CALL MFNM (NC,AM2,AMF2)
   CALL MFNM (NC,AMR,AMFR)
   AMOLR = 1.0 - (AMOL1 + AMOL2)

   GO TO 90

60 DO 70, I = 1, NC

      AIV(I) = ALOG(AMFR(I) / AMF1(I))
      AIV(NC+I) = ALOG(AMFR(I) / AMF2(I))

70 CONTINUE

   IF (IABS(IPARM(4)).EQ.2) THEN
```

```
            CALL ZSCNT (FFCN3,NSIG,NC2,MAXIT,OMFR,AIV,FNM,WK,IER)
            IF (IER.NE.0) IER = 220

         ELSE

            CALL ZSPOW (FFCN3,NSIG,NC2,MAXIT,OMFR,AIV,FNM,WK,IER)
            IF (IER.NE.0) IER = 230

         ENDIF
      IF (IER.NE.0) GO TO 100

      DO 80, I = 1, NC

         AM1(I) = EXP(AIV(I))
         AM2(I) = EXP(AIV(NC+I))

   80 CONTINUE

      CALL PDIST3 (NC,OMFR,AM1,AM2)
      CALL MSSCN (NC,OMFR,AM1,AM2,AMF1,AMF2,AMFR)

   90 IPARM(24) = NFE
      RPARM(6) = FNM

  100 RETURN
      END
      SUBROUTINE GFCN3 (AIV,GON,NC2,OMFR)
      COMMON /AAAA2/ NFE, AMN(20), AMF1(20), AMF2(20), AMFR(20),
     1               WRK5(20)
      COMMON /AAAA7/ FGC1(20), FGC2(20), FGCR(20)
      DIMENSION AIV(1), GON(1), OMFR(1)

      NC = NC2 / 2
      CALL CKMN3 (NC,AIV,OMFR)
      DO 10, I = 1, NC
         AMN(I) = AIV(I)
   10 CONTINUE

      CALL MFNM (NC,AMN,AMF1)
      CALL FGCF (NC,AMF1,FGC1,AMN,Z)
      DO 20, I = 1, NC
         AMN(I) = AIV(NC+I)
   20 CONTINUE
      CALL MFNM (NC,AMN,AMF2)
      CALL FGCF (NC,AMF2,FGC2,AMN,Z)

      DO 30, I = 1, NC
         AMN(I) = OMFR(I) - (AIV(I) + AIV(NC+I))
   30 CONTINUE
      CALL MFNM (NC,AMN,AMFR)
      CALL FGCF (NC,AMFR,FGCR,AMN,Z)
```

```
      DO 40, I = 1, NC

         GON(I) = ALOG(AMF1(I) * FGC1(I)) - ALOG(AMFR(I) * FGCR(I))
         GON(NC+I) = ALOG(AMF2(I) * FGC2(I)) - ALOG(AMFR(I) * FGCR(I))

   40 CONTINUE

      NFE = NFE + 1

      RETURN
      END
      SUBROUTINE FFCN3 (AIV,FOK,NC2,OMFR)
      COMMON /AAAA2/ NFE, EQR1(20), EQR2(20), AMF1(20), AMF2(20),
     1               AMFR(20)
      COMMON /AAAA7/ FGC1(20), FGC2(20), FGCR(20)
      DIMENSION AIV(1), FOK(1), OMFR(1)

      NC = NC2 / 2

      DO 10, I = 1, NC

         EQR1(I) = EXP(AIV(I))
         EQR2(I) = EXP(AIV(NC+I))

   10 CONTINUE

      CALL PDIST3 (NC,OMFR,EQR1,EQR2)
      CALL MSSCN (NC,OMFR,EQR1,EQR2,AMF1,AMF2,AMFR)
      CALL FGCF (NC,AMF1,FGC1,EQR1,Z)
      CALL FGCF (NC,AMF2,FGC2,EQR1,Z)
      CALL FGCF (NC,AMFR,FGCR,EQR1,Z)

      DO 20, I = 1, NC

         FOK(I) = ALOG(AMF1(I) * FGC1(I)) - ALOG(AMFR(I) * FGCR(I))
         FOK(NC+I) = ALOG(AMF2(I) * FGC2(I)) - ALOG(AMFR(I) * FGCR(I))
   20 CONTINUE

      NFE = NFE + 1

      RETURN
      END
      SUBROUTINE CKMN3 (NC,AMN,OMFR)
      DIMENSION AMN(1), OMFR(1)

      DO 10, I = 1, NC

         IF (AMN(I).LE.0.0) GO TO 20
         IF (AMN(NC+I).LE.0.0) GO TO 20
         DIFF = OMFR(I) - (AMN(I) + AMN(NC+I))
```

```
         IF (DIFF.LE.0.0) GO TO 20

   10 CONTINUE
      RETURN

   20 IER = 301
      CALL ERROPR (IER)

      RETURN
      END
      SUBROUTINE WRT4 (NC,AMF1,AMF2,AMFR)
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      DIMENSION AMF1(1), AMF2(1), AMFR(1)

      WRITE (7,100)
      WRITE (7,110) AMOL1, AMOL2, AMOLR
      WRITE (7,120)
      WRITE (7,130) (I,AMF1(I),AMF2(I),AMFR(I), I = 1, NC)

      RETURN

  100 FORMAT (//,1X,*STARTING INFORMATION FOR THREE PHASE FLASH*)
  110 FORMAT (/,1X,*PHASE MOLES, 1*,E25.10,
     1   /,1X,*PHASE MOLES, 2*,E25.10,/,1X,*PHASE MOLES, R*,E25.10)
  120 FORMAT (/,2X,9HCOMPONENT,13X,7HPHASE 1,13X,7HPHASE 2,
     1   13X,7HPHASE R,/,6X,5HINDEX,3(9X,11HCOMPOSITION),
     2   /,1X,10H---------- ,3(5X,15H---------------),/)
  130 FORMAT (1X,I10,3E20.12)

      END
      SUBROUTINE WRT5 (NC,AMF1,AMFR)
      COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
      DIMENSION AMF1(1), AMFR(1)

      WRITE (7,100)
      WRITE (7,110) AMOL1, AMOLR
      WRITE (7,120)
      WRITE (7,130) (I,AMF1(I),AMFR(I), I = 1, NC)

      RETURN

  100 FORMAT (//,1X,*FINAL SOLUTION FROM TWO-PHASE SECTION*)
  110 FORMAT (/,1X,*PHASE MOLES, 1*,5X,E20.10,
     1   /,1X,*PHASE MOLES, R*,5X,E20.10)
  120 FORMAT (/,2X,9HCOMPONENT,13X,7HPHASE 1,13X,7HPHASE R,
     1   /,6X,5HINDEX,2(9X,11HCOMPOSITION),/,1X,10H---------- ,
     2   2(5X,15H--------------- ),//)
  130 FORMAT (1X,I10,2E20.12)

      END
      SUBROUTINE EXPHCK (NC,AMF1,AMF2,AMFR,FGC1,FGC2,FGCR,IEP)
```

```
   COMMON /AAAA1/ IPARM(26), EPS(8), RPARM(6), AMOL1, AMOL2, AMOLR
   COMMON /AAAA7/ EQR1N(20), EQR2N(20), WK(20)
   COMMON /AAA10/ IPCFC, RESN, PFGC(20)
   DIMENSION AMF1(1), AMF2(1), AMFR(1)
   DIMENSION FGC1(1), FGC2(1), FGCR(1)

   IEOS = IPARM(1)

   IF (IPCFC.NE.0) GO TO 10
   CALL PCFGC (NC,IEOS,PFGC)
   IPCFC = 1

10 TFEM3P = 0.0
   DO 20, I = 1, NC

      DGP1 = AMF1(I) * ALOG(AMF1(I) * FGC1(I) / PFGC(I))
      DGP2 = AMF2(I) * ALOG(AMF2(I) * FGC2(I) / PFGC(I))
      DGPR = AMFR(I) * ALOG(AMFR(I) * FGCR(I) / PFGC(I))
      TFEM3P = TFEM3P + AMOL1 * DGP1 + AMOL2 * DGP2 + AMOLR * DGPR

20 CONTINUE

   IF ((AMOL2.LT.AMOL1).OR.(AMOL2.LT.AMOLR)) THEN

           DO 30, I = 1, NC
              FGC1(I) = AMF2(I)
30         CONTINUE
           AMN1 = AMOL2
           IF (AMOL1.LT.AMOLR) THEN

                  DO 40, I = 1, NC

                     FGC2(I) = AMF1(I)
                     FGCR(I) = AMFR(I)

40                CONTINUE
                  AMN2 = AMOL1
                  AMN3 = AMOLR

               ELSE

                  DO 50, I = 1, NC

                     FGC2(I) = AMFR(I)
                     FGCR(I) = AMF1(I)

50                CONTINUE
                  AMN2 = AMOLR
                  AMN3 = AMOL1

               ENDIF
```

```
               ELSE

                  DO 60, I = 1, NC

                     FGC1(I) = AMF1(I)
                     FGC2(I) = AMFR(I)
                     FGCR(I) = AMF2(I)

60                CONTINUE
                  AMN1 = AMOL1
                  AMN2 = AMOLR
                  AMN3 = AMOL2

               ENDIF

      DO 70, I = 1, NC

         A1 = AMN1 * FGC1(I)
         A2 = AMN2 * FGC2(I)
         A3 = AMN3 * FGCR(I)
         FGC1(I) = A1 + A1 * A3 / (A1 + A2)
         FGC2(I) = A2 + A2 * A3 / (A1 + A2)

70    CONTINUE

      AMN1 = 0.0
      DO 80, I = 1, NC

         AMN1 = AMN1 + FGC1(I)
         FGCR(I) = FGC1(I)
         WK(I) = FGC2(I)

80    CONTINUE

      AMN2 = 1.0 - AMN1
      CALL MFNM (NC,FGCR,FGC1)
      CALL MFNM (NC,WK,FGC2)

      CALL FGCF (NC,FGC1,FGCR,WK,Z)
      DGP1 = 0.0
      DO 90, I = 1, NC
         DGP1 = DGP1 + FGC1(I) * ALOG(FGC1(I) * FGCR(I) / PFGC(I))
90    CONTINUE

      CALL FGCF(NC,FGC2,FGCR,WK,Z)
      DGP2 = 0.0
      DO 100, I = 1, NC
         DGP2 = DGP2 + FGC2(I) * ALOG(FGC2(I) * FGCR(I) / PFGC(I))
100   CONTINUE
```

```
      TFEM2P = AMN1 * DGP1 + AMN2 * DGP2
      DIFFG = TFEM3P - TFEM2P
      IF (DIFFG.GT.0.0) GO TO 110
      IEP = 0
      RETURN

  110 DO 120, I = 1, NC

         AMF1(I) = FGC1(I)
         AMFR(I) = FGC2(I)

  120 CONTINUE
      AMOL1 = AMN1
      AMOLR = AMN2
      IEP = 1

      RETURN
      END
      SUBROUTINE PCFGC (NC,IEOS,PFGC)
      COMMON /AAAA3/ TEM, PRS
      COMMON /AAAA5/ AI(20), BI(20), AIJ(20,20)
      DIMENSION PFGC(1)

      R = 10.732

      DO 40, I = 1, NC

         BB = BI(I) * PRS / (TEM * R)
         AA = AI(I) * PRS / (TEM * R)**2

         IF (IEOS.EQ.1) THEN

                A2 = BB - 1.0
                A1 = AA - BB * (3.0 * BB + 2.0)
                A0 = BB * (BB * (BB + 1.0) - AA)

             ELSE

                A2 = -1.0
                A1 = AA - BB * (1.0 + BB)
                A0 = -AA * BB

             ENDIF

         CALL ZCALC (A2,A1,A0,Z1,Z2,Z3,IJZ)

         IF (IJZ.LT.0) GO TO 30
         IF (IJZ.EQ.3) GO TO 10
         IF (Z1.LE.0.0) GO TO 30
         IF (Z2.LE.0.0) GO TO 30
         ZL = AMIN1(Z1,Z2)
```

```
          ZH = AMAX1(Z1,Z2)
          GO TO 20

    10    ZH = AMAX1(Z1,Z2,Z3)
          ZL = ZH
          IF ((Z1.GT.0.0).AND.(Z1.LE.ZL)) ZL = Z1
          IF ((Z2.GT.0.0).AND.(Z2.LE.ZL)) ZL = Z2
          IF ((Z3.GT.0.0).AND.(Z3.LE.ZL)) ZL = Z3

    20    CALL PSUBPHI (AA,BB,IEOS,ZH,PFGCH)
          CALL PSUBPHI (AA,BB,IEOS,ZL,PFGCL)

          DIFF = PFGCH - PFGCL
          IF (DIFF.LT.0.0) THEN
                 PFGC(I) = EXP(PFGCH)
              ELSE
                 PFGC(I) = EXP(PFGCL)
              ENDIF
          GO TO 40

    30    ZH = AMAX1(Z1,Z2)
          CALL PSUBPHI (AA,BB,IEOS,ZH,PFGCH)
          PFGC(I) = EXP(PFGCH)

    40 CONTINUE

       RETURN
       END
       SUBROUTINE PSUBPHI (A,B,IEOS,Z,FGC)

       IF (IEOS.NE.1) GO TO 10

       C1 = 1.0 + SQRT(2.0)
       C2 = 1.0 - SQRT(2.0)
       AJ = A / (2.0 * B * SQRT(2.0))
       AK = (Z + C1 * B) / (Z + C2 * B)
       GO TO 20

    10 AJ = A / B
       AK = (Z + B) / Z

    20 FGC = Z - 1.0 - ALOG(Z - B) - AJ * ALOG(AK)

       RETURN
       END
```

# BIBLIOGRAPHY

[B1] Baker, L. E., Pierce, A. C., and Luks, K. D.: "Gibbs Energy Analysis of Phase Equilibria," Soc. Pet. Eng. J. (October 1982).

[E1] Edmister, W. C.: "Applied Hydrocarbon Thermodynamics," Petroleum Refiner (April 1958).

[F1] Fussell, D. D., and Yanosik, J. L.: "An Iterative Sequence for Phase Equilibria Calculations Incorporating the Redlich-Kwong Equation of State," Soc. Pet. Eng. J. (June 1978).

[F2] Fussell, L. T.: "A Technique for Calculating Multiphase Equilibria," Soc. Pet. Eng. J. (August 1979).

[G1] Gardner, J. W., Orr, F. M., and Patel, P. D.: "The Effect of Phase Behavior on $CO_2$ Flood Displacement Efficiency," SPE 8367, presented at the 54th Annual Fall Conference and Exhibition of SPE - AIME, Las Vegas, Nevada, Sept. 23-26, 1979.

[I1] Im, U. K., and Kurata, F.: "Heterageneous Phase Behavior of Carbon Dioxide in n-Hexane and n-Heptane at Low Temperatures," J. Chem. and Eng. Data (October 1971).

[I2] International Mathematical and Statistical Library Reference Manual, Edition 9, IMSL Inc. (1982), NBC Building, 7500 Bellaire Blvd., Houston, TX 77036.

[K1] Kulkarni, A. A., Zarah, B. Y., Luks, K. D., and Kohn, J. P.: "Phase-Equilibria Behavior of System Carbon Dioxide-n-Decane at Low Temperatures," J. Chem. and Eng. Data (January 1974).

[L1] Lawal, A.: "New Convergent Algorithms for Compositional Reservoir Models," Ph.D. dissertation, The University of Texas at Austin (1984).

[M1] Mehra, R. K.: "Computation of Multi-Phase Equilibrium in Compositional Reservoir Studies," Ph.D. dissertation, Univ. of Calgary (1981).

[M2] Mehra, R. K., Heidemann, R. A., and Aziz, K. A.: "Acceleration of the Successive Substitution Method of Vapor-Liquid Equilibrium Calculations," paper presented at the 30th Cdn. Chem. Eng. Conf., Edmonton, Oct. 20-22, 1980.

[M3] Mehra, R. K., Heidemann, R. A., and Aziz, K. A.: "Computation of Multiphase Equilibrium for Compositional Simulation," Soc. Pet. Eng. J. (February 1982).

[M4] Meldrum, A. H., and Nielsen, R. F.: "A Study of Three-Phase Equilibria for Carbon Dioxide-Hydrocarbon Mixtures," Prod. Monthly (August 1955).

[M5] Metcalfe, R. A., and Yarborough, L.: "Effect of Phase Equilibria on the $CO_2$ Displacement Mechanism," SPE 7061, presented at the SPE-AIME Symposium on Improved Oil Recovery, Tulsa, April 16-19, 1978.

[M6] Michelsen, M. L.: "The Isothermal Flash Problem. Part I. Stability," Fluid Phase Equilibria, 9 (1982).

[M7] Michelsen, M. L.: "The Isothermal Flash Problem. Part II. Phase-Split Calculation," Fluid Phase Equilibria, 9 (1982).

[M8] Modell, M., and Reid, R. C.: Thermodynamics and its Applications, Prentice-Hall (1974).

[M9] Moré, J., Garbow, B., and Hillstrom, K.: User Guide for MINPACK—1, Argonne National Laboratory Report ANL-80-74, Argonne, Illinois, August (1980).

[N1] Nagy, Z., and Shirkovskiy, A. I.: "Mathematical Simulation of Natural Gas Condensation Processes Using the Peng-Robinson Equation of State," SPE 10982, presented at the 57th Annual Fall Technical Conference and Exhibition of SPE-AIME, New Orleans, LA, Sept. 26-29, 1982.

[N2] Nghiem, L. X., and Aziz, K. A.: "A Robust Iterative Method for Flash Calculations Using the Soave-Redlich Kwong or Peng-Robinson Equation of State," SPE 8285, presented at the 54th Annual Fall Technical Conference and Exhibition of SPE-AIME, Las Vegas, Nevada, Sept. 23-26, 1979.

[N3] Nghiem, L. X., and Li, Y. K.: "Effect of Phase Behavior on $CO_2$ Displacement Efficiency at Low Temperatures: Model Studies With an Equation of State," SPE 13116, presented at the 59th Annual Fall Technical Conference and Exhibition of SPE-AIME, Houston, TX, Sept. 16-19, 1984.

[O1] Olds, R. H., Reamer, H. H., Sage, B. H., and Lacey, W. N.: "Phase Equilibria in Hydrocarbon Systems," Ind. and Eng. Chem. (March 1949).

[O2] Orr, F. M., Jr., and Jensen, C. M.: "Interpretation of Pressure-Composition Phase Diagrams for $CO_2$/Crude-Oil Systems," SPE 1125, presented at the 57th Annual Fall Technical Conference and Exhibition of SPE-AIME, New Orleans, LA, Sept. 26-29, 1982.

[O3] Orr, F. M., Jr., Lien, C. L., and Pelletier, M. T.: "Liquid-Liquid Phase Behavior in $CO_2$-Hydrocarbon Systems," Symposium on chemistry of Enhanced Oil Recovery, American Chemical Society, Atlanta, March 29—April 3, 1981.

[O4] Orr, F. M., Jr., Yu, A. D., and Lien, C. L.: "Phase Behavior of $CO_2$ and Crude Oil in Low Temperature Reservoirs," SPE 8813, presented at the First SPE/DOE Symposium on Enhanced Oil Recovery, Tulsa, OK, April 20-23, 1980.

[P1] Peng, D. Y., and Robinson, D. B.: "A New Two-Constant Equation of State," Ind. and Eng. Chem. Fundamentals (February 1976).

[P2] Poettmann, F. H., and Katz, D. L.: "Phase Behavior of Binary Carbon Dioxide—Paraffin Systems," Ind. and Eng. Chem. (September 1945).

[P3] Powell, M. J. D.: "A Hybrid Method for Nonlinear Equations," Numerical Methods for Nonlinear Algebraic Equations, P. Rabinowitz, Editor, Gordon and Breach, London (1970).

[R1] Reamer, H. H., and Sage, B. H.: "Phase Equilibria in Hydrocarbon Systems. Volumetric and Phase Behavior in the n-Decane-$CO_2$ System," J. Chem. and Eng. Data (October 1963).

[R2] Redlich, O., and Kwong, J. N. S.: "On the Thermodynamics of Solutions. V--An Equation of State. Fugacities of Gaseous Solutions," Chem. Reviews (February 1949).

[R3] Riazi, M. R.: "Prediction of Thermophysical Properties of Petroleum Fractions," Ph.D. dissertation, Penn State University (1979).

[R4] Risnes, R., and Dalen, U.: "Equilibrium calculations for Coexisting Liquid Phases," SPE 11126, presented at the 57th Annual Fall Technical Conference and Exhibition of SPE-AIME, New Orleans, LA, Sept. 26-29, 1982.

[S1] Shelton, J. L., and Yarborough, L.: "Multiple Phase Behavior in Porous Media During $CO_2$ or Rich-Gas Flooding," J. Pet. Tech. (Sept. 1977).

[S2] Simon, R., Rosman, A., and Zana, E.: "Phase Behavior Properties of $CO_2$-Reservoir Oil Systems," Soc. Pet. Eng. J. (February 1978).

[S3] Soave, G.: "Equilibrium Constants From a Modified Redlich-Kwong Equation of State," Chem. Eng. Sci. (June 1972).

[S4] Stewart, W. C., and Nielsen, R. F.: "Phase Equilibria for Mixtures of Carbon Dioxide and Several Normal Saturated Hydrocarbons," Prod. Monthly (Jan. 1954).

[T1] Thele, K. J.: "A Comparison of Three Equation of State Compositional Simulators," M.S. thesis, The University of Texas at Austin (1984).

[T2] Turek, E. A., Metcalfe, R. S., and Fishback, R. E.: "Phase Behavior of Several $CO_2$-West Texas Reservoir Oil Systems," SPE 13117, presented at the 59th Annual Fall Technical Conference and Exhibition, Houston, TX, Sept. 16-19, 1984.

[W1] Wilson, G. M.: "A Modified Redlich-Kwong Equation of State--Application to General Physical Data Calculations," 65th National AIChE Meeting, Cleveland, OH, May 4-7, 1969.

[W2] Wolfe, P.: "The Secant Method for Simultaneous Non-Linear Equations," Communications of the ACM, 2 (1959).

# VITA

Douglas Roger Perschke was born February 20, 1959, in Rapid City, South Dakota, the son of Roger W. Perschke and Ruby L. Perschke. After completing high school in Missoula, Montana, he attended the University of Oregon in Eugene, Oregon. After one year of study at Oregon, he transferred to the University of Washington in Seattle, Washington, where he received a Bachelor of Science degree in Chemistry in May of 1981. He entered The University of Texas at Austin in August of 1981.

Permanent address: 11500 Braesview #2806
San Antonio, TX 78213

This thesis was typed by Speedway Typing.