MENTATION PAGE

Form Approved
OMB No. 0704-0188

is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources,
ing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this
ing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson
[illegible], and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
May 1992

3. REPORT TYPE AND DATES COVERED
M.S. Thesis

4. TITLE AND SUBTITLE
A Generalized Polynomial Form of the Objective Function in Flash Calculations

5. FUNDING NUMBERS

6. AUTHOR(S)
Brett D. Weigle, Captain, U.S. Army

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Petroleum and Natural Gas Engineering
207 Mineral Science Building
Pennsylvania State University
University Park, PA 16802

8. PERFORMING ORGANIZATION REPORT NUMBER

DTIC
ELECTE
APR 0 2 1992
S D
D

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Advanced Civil Schooling Program
Quartermaster Branch
U.S. Army

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Prepared in partial fulfillment of the requirements for the degree of Master of Science in Petroleum and Natural Gas Engineering

12a. DISTRIBUTION/AVAILABILITY STATEMENT
unlimited

This document has been approved for public release and sale; its distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

This work centers on the recasting of the Rachford-Rice objective function into a polynomial function of the vapor fraction. The degree of this polynomial is one less than the number of components in the system and its coefficients can be calculated from the feed composition and the equilibrium ratios. A recursive expression is developed that involves symmetric functions and can be easily programmed on a computer or scientific calculator. The principal advantage of this new form of the objective function is that the theory of polynomials is well-developed. The location of their zeroes can be predicted with confidence by techniques based on sound mathematical principles, such as the Fourier-Budan theorem. The polynomial in the vapor fraction is well-behaved over the two-phase region and its root can be quickly located by a hybrid method of interval-halving technique and Newton-Raphson procedure. The validity of the new objective function and its automatic coefficent-generating algorithm are tested using several multicomponent systems for which experimental data are available. The new objective function is not prone to the erratic behavior of the Rachford-Rice function and is not sensitive to initial guess of the root.

14. SUBJECT TERMS
polynomial; objective function; flash calculations; vapor-liquid equilibrium; phase behavior; equilibrium ratio; equation of state; symmetric functions

15. NUMBER OF PAGES
117

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UNLIMITED

GENERAL INSTRUCTIONS FOR COMPLETING SF 298

The Report Documentation Page (RDP) is used in announcing and cataloging reports. It is important that this information be consistent with the rest of the report, particularly the cover and title page. Instructions for filling in each block of the form follow. It is important to **stay within the lines to meet optical scanning requirements.**

Block 1. Agency Use Only (Leave Blank)

Block 2. Report Date. Full publication date including day, month, and year, if available (e.g. 1 Jan 88). Must cite at least the year.

Block 3. Type of Report and Dates Covered. State whether report is interim, final, etc. If applicable, enter inclusive report dates (e.g. 10 Jun 87 - 30 Jun 88).

Block 4. Title and Subtitle. A title is taken from the part of the report that provides the most meaningful and complete information. When a report is prepared in more than one volume, repeat the primary title, add volume number, and include subtitle for the specific volume. On classified documents enter the title classification in parentheses.

Block 5. Funding Numbers. To include contract and grant numbers; may include program element number(s), project number(s), task number(s), and work unit number(s). Use the following labels:

C	Contract	**PR**	Project
G	Grant	**TA**	Task
PE	Program Element	**WU**	Work Unit Accession No.

Block 6. Author(s). Name(s) of person(s) responsible for writing the report, performing the research, or credited with the content of the report. If editor or compiler, this should follow the name(s).

Block 7. Performing Organization Name(s) and Address(es). Self-explanatory.

Block 8. Performing Organization Report Number. Enter the unique alphanumeric report number(s) assigned by the organization performing the report.

Block 9. Sponsoring/Monitoring Agency Names(s) and Address(es). Self-explanatory.

Block 10. Sponsoring/Monitoring Agency. Report Number. (If known)

Block 11. Supplementary Notes. Enter information not included elsewhere such as: Prepared in cooperation with...; Trans. of ..., To be published in .... When a report is revised, include a statement whether the new report supersedes or supplements the older report.

Block 12a. Distribution/Availablity Statement. Denote public availability or limitation. Cite any availability to the public. Enter additional limitations or special markings in all capitals (e.g. NOFORN, REL, ITAR)

DOD	See DoDD 5230.24, "Distribution Statements on Technical Documents."
DOE	See authorities
NASA	See Handbook NHB 2200.2.
NTIS	Leave blank.

Block 12b. Distribution Code.

DOD	DOD - Leave blank
DOE	DOE - Enter DOE distribution categories from the Standard Distribution for Unclassified Scientific and Technical Reports
NASA	NASA - Leave blank
NTIS	NTIS - Leave blank.

Block 13. Abstract. Include a brief (Maximum 200 words) factual summary of the most significant information contained in the report.

Block 14. Subject Terms. Keywords or phrases identifying major subjects in the report.

Block 15. Number of Pages. Enter the total number of pages.

Block 16. Price Code. Enter appropriate price code (NTIS only).

Blocks 17. - 19. Security Classifications. Self-explanatory. Enter U.S. Security Classification in accordance with U.S. Security Regulations (i.e., UNCLASSIFIED). If form contains classified information, stamp classification on the top and bottom of the page.

Block 20. Limitation of Abstract. This block must be completed to assign a limitation to the abstract. Enter either UL (unlimited) or SAR (same as report). An entry in this block is necessary if the abstract is to be limited. If blank, the abstract is assumed to be unlimited.

The Pennsylvania State University
The Graduate School
Department of Mineral Engineering

A GENERALIZED POLYNOMIAL FORM OF THE OBJECTIVE FUNCTION IN FLASH CALCULATIONS

A Thesis in

Petroleum and Natural Gas Engineering

by

Brett D. Weigle

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 1992

Brett D. Weigle

Brett D. Weigle

We approve the thesis of Brett D. Weigle.

Date of Signature

Michael Adewumi
12/18/91
Michael A. Adewumi
Associate Professor of Petroleum
and Natural Gas Engineering
Thesis Advisor

12/18/91
Turgay Ertekin
Professor of Petroleum and
Natural Gas Engineering
Section Chair of Petroleum and
Natural Gas Engineering

Paul J. Hicks
12-18-91
Paul J. Hicks, Jr.
Assistant Professor of Petroleum
and Natural Gas Engineering

David M. Bressoud
12/18/91
David M. Bressoud
Professor of Mathematics

ABSTRACT

The Rachford-Rice objective function for flash calculations exhibits a nearly flat slope across the two-phase region and sharp discontinuities near the dewpoint. These features make iterative solution procedures sensitive to the initial estimate of the root and prone to spurious values if a correction step throws the algorithm outside the two-phase region or near the phase boundary.

This work centers on the recasting of the Rachford-Rice objective function into a polynomial function of the vapor fraction, α. The degree of this polynomial is one less than the number of components in the system and its coefficients can be calculated from the feed composition and the equilibrium ratios. A recursive expression is developed that involves symmetric functions and can be easily programmed on a computer or scientific calculator.

The principal advantage of this new form of the objective function is that the theory of polynomials is well-developed. The location of their zeros can be predicted with confidence by techniques based on sound mathematical principles, such as the Fourier-Budan theorem. The α-polynomial is well-behaved over the two-phase region and its root can be quickly located by a hybrid method of interval-halving technique and Newton-Raphson procedure. The validity of the new objective function and its automatic coefficient-generating algorithm are tested using several multicomponent systems for which experimental data are available.

Results of these tests prove conclusively the validity of the generalized polynomial objective function. The versatility of this form of the flash objective function, compared with the original Rachford-Rice version, is demonstrated. Another potential advantage of the polynomial form is its ability to handle dilute systems in which some components are present but in very low concentrations. It also promises possible usage as a means of developing appropriate lumping schemes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Roman

A = pressure-dependent constant [Equation (2.6)]
A = parameter for the PREOS [Equation (2.18)]
A = number of sign changes in derivative series (Section 5.2.4)
a = intercept of log Kp vs F plot [Equation (2.3)]
a = unitary or elementary symmetric function [Equation (3.41)]
$a(T)$ = attractive constant for PREOS [Equation (2.11)]
$a(T_c)$ = attractive constant at critical point for PREOS [Equation (2.10)]
B = parameter for the PREOS [Equation (2.19)]
B = number of sign changes in derivative series (Section 5.2.4)
b = translation constant for Cox chart [Equation (2.1)]
b = molar co-volume for PREOS [Equation (2.9)]
C_i = constant used in the objective function [Equation (3.16)]
$\hat{C}_i$ = excluded term [Equation (3.42)]
c = slope of log Kp vs F plot [Equation (2.3)]
c = volume translation parameter [Equation (2.21)]
F = number of moles in feed stream (Section 3.1.1)
F = component characterization factor [Equation (2.1)]
K_i = equilibrium ratio (Section 1.2)
L = number of moles in the liquid phase (Section 3.1.1)
$\mathbf{M}$ = interexpressibility matrix [Equation (4.12)]
m = element of matrix $\mathbf{M}$ [Equation (4.31)]
N = number of components in the fluid system (Section 1.3)

n = number of terms in the set [Equation (3.41)]

P = a partition of r into at most λ parts [Equation (4.8)]

p = pressure

p_i = part of a partition [Equation (4.1)]

$p(t)$ = polynomial as a function of t (Section 5.2.4)

$p(u)$ = polynomial as a function of u (Section 5.2.4)

$p(v)$ = polynomial as a function of v (Section 5.2.4)

R = real gas constant [Equation (2.8)]

$\mathbf{R}$ = field of real numbers (Section 5.2.4)

s = power sum symmetric function [Equation (4.6)]

s_P = slope of plot [Equation (2.2)]

T = temperature

t = exponential term in generating function [Equation (3.41)]

u = augmented unitary symmetric function [Equation (4.7)]

u = real number (Section 5.2.4)

V = number of moles in the vapor phase (Section 3.1.1)

v = molar volume

$\tilde{v}$ = pseudo volume [Equation (2.21)]

v = length of a partition [Equation (4.2)]

v = real number (Section 5.2.4)

w = weight of a partition [Equation (4.1)]

X = fluid "map" coordinate from Varotsis (Section 2.2)

X_i = component "map" coordinate from Varotsis (Section 2.2)

x = mole fraction in the liquid phase (Section 3.1.1)

x = argument of symmetric function [Equation (3.41)]

Y = fluid "map" coordinate from Varotsis (Section 2.2)

Y_i = component "map" coordinate from Varotsis (Section 2.2)

y = mole fraction in the vapor phase (Section 3.1.1)

Z = compressibility factor [Equations (2.20), (2.22)]

z = mole fraction in the feed stream (Section 3.1.1)

Greek

α = vapor fraction [Equation (1.1)]

β = isothermal compressibility (Section 2.3.2)

δ = binary interaction coefficient [Equation (2.16)]

δ = bulk modulus [Equation (2.23)]

η = function to describe $u(T)$ away from critical point [Equation (2.11)]

κ = function of acentric factor in PREOS [Equation (2.13)]

κ_0 = function of acentric factor in PRSV EOS [Equation (2.26)]

κ_1 = parameter in PRSV EOS [Equation (2.27)]

μ = coefficient of polynomial [Equation (5.2)]

π_i = multiplicity of a part in a partition [Equation (4.1)]

ψ = EOS variable [Equations (2.24), (2.25)]]

ω_i = Pitzer acentric factor for the i-th component [Equation (2.5)]

Subscripts

B = boiling point [Equation (2.1)]

c = critical property

i,j,k,l,m = individual components of the fluid system

ij = interaction between component i and component j of the fluid system

k = convergence pressure (Section 2.2)

P = constant pressure [Equation (4.29)]

R = reduced property

RA = Rackett compressibility factor [Equation (2.22)]

r = order of symmetric function [Equation (3.41)]

T = constant temperature [Section 2.3.2, Equation (4.29)]

Superscripts

I = fluid state [Equations (2.24), (2.25)]

i = component of fluid system [Equations (2.24), (2.25)]

(n) = order of derivative

$'$ = first derivative

$''$ = second derivative

$'''$ = third derivative

Abbreviations

CPU = computer central processing unit

EOS = equation of state

°F = degree Fahrenheit

LHS = left-hand side (of an equation)

PREOS = Peng-Robinson equation of state

PRSV EOS = Peng-Robinson-Stryjek-Vera equation of state

psia = pounds per square inch, absolute

Q.E.D. = *quod erat demonstrandum*, which was to be proved (Appendix A)

RHS = right-hand side (of an equation)

SRKEOS = Soave-Redlich-Kwong equation of state

SSM = successive substitution method

ACKNOWLEDGMENTS

The author acknowledges the United States Army for affording him the opportunity to study for this degree.

The author expresses gratitude to his thesis advisor, Dr. Michael A. Adewumi, whose enthusiasm was a driving force behind this research. His keen grasp of the physics of phase behavior and his mathematical expertise were invaluable in seeing it to its completion.

The author wishes to thank Captain John H. Warren, United States Army, for his pioneering work in this field. Without the foundation provided by his thesis, this work would not have reached conception, much less fruition.

The author expresses appreciation to Dr. David M. Bressoud for his lessons in combinatory analysis and symmetric functions. He provided the clue that finally yielded the recursive form of the α-polynomial.

The author offers his appreciation to the other members of his thesis committee, Dr. Turgay Ertekin and Dr. Paul J. Hicks, Jr., for their time and comments in evaluating this work.

Mr. Timothy Kohler is the source of most of the author's ability to efficiently utilize the department's computers and was always there to answer a question or deal with a recalcitrant machine.

The author extends his gratitude to his wife, Deborah Lee Weigle, for cheerfully tolerating a work schedule worse than the one imposed on the author as a company commander. Her support and understanding were instrumental in the completion of this thesis.

Chapter 1

DISCUSSION OF THE PROBLEM

1.1 Introduction

Determination of the equilibrium state of coexisting liquid and vapor phases, particularly for multicomponent fluid mixtures, is of vital interest to the petroleum and chemical industries. Many processes in petroleum production and refining involve repetitive flash calculations for design and operational purposes. The primary goal of performing flash calculations is to determine the relative amounts and compositions of the coexisting phases for a given feed composition at a specified condition of temperature and pressure.

This work is confined solely to two-phase vapor-liquid equilibrium computations, although its results will no doubt find application in multiphase flash problems in the future.

1.2 The Generic Flash Algorithm

To begin the calculation, the following variables must be specified: the system pressure and temperature, the molar composition of the feed stream, z_i , and an initial estimate of the equilibrium ratios, $K_i = \frac{y_i}{x_i}$. The process is assumed to occur under isothermal and isobaric conditions. The stages of the calculation are:

1. Compute initial estimates of the equilibrium ratios by one of the established techniques or by an empirical correlation.

2. Calculate the phase distribution and compositions corresponding to the given K-values. This involves the iterative solution of the following objective function, developed by Rachford and Rice (1952):

$$\sum_{i=1}^{N} \frac{z_i \, (K_i - 1)}{1 + \alpha \, (K_i - 1)} = 0 \tag{1.1}$$

where α is the vapor molar fraction.

3. Use an equation of state (EOS) to calculate the component fugacities in each phase and check for equality.

4. If equality is not achieved (i.e., the phases are not in equilibrium), correct the K-values on the basis of the fugacities and repeat steps 2-4.

This correction is commonly performed using a successive substitution-type method or a second-order Newton-type scheme. These algorithms are well-known and are described in several papers [e.g., Risnes et al. (1981); Michelsen (1982); Boston and Britt (1978)].

Successful implementation of the generic flash algorithm described above requires three principal elements. These are (1) a general estimate of the set of equilibrium ratios to start the procedure; (2) a good equation of state to improve K_i; and (3) a robust objective function that guarantees convergence to a single value of α. A poor first guess of K-values may produce a phase split that is physically impossible under the prevailing pressure and temperature. Satisfactory methods are available for generating these values. Furthermore, existing equations of state do a fairly good job of predicting phase properties, and other efforts continue along this line.

One area that has not enjoyed equal amounts of attention for a long time is the form of the objective function. Invariably, the Rachford-Rice objective function [Equation (1.1)] is most often used. Recent investigations (Warren, 1991) have shown that this objective function does exhibit some strange behavior which may affect its ability to generate good results for some conditions.

1.3 Objectives of the Investigation

This study is aimed at evolving a generalization of the new polynomial form of the Rachford-Rice objective function developed by Warren (1991). The resulting generalized polynomial function of the vapor fraction, α, should be capable of handling an N-component mixture. The coefficients of the generalized polynomial should depend on only two variables, the molar composition of the feed stream and the equilibrium ratios, and should be easy to obtain, either analytically or numerically. Appropriate algorithms are to be developed for this purpose.

The principal advantage of a polynomial form of the flash-calculation objective function is that the theory of polynomials is well-developed and semi-analytical solution techniques exist for equations up to fifth-order (Zaguskin, 1961). For higher-order polynomials, the Newton-Raphson iterative method usually provides a fast and accurate determination of the roots.

Determination of all the zeros of this polynomial is unnecessary since the physics of the problem demands that only the zeros on the bounded interval [0,1] are of practical interest. Furthermore, the physics also suggests that only one zero (or value of α) exists on this interval, which represents the two-phase vapor-liquid region. It can be shown mathematically that this is indeed the case for well-defined systems, as will be demonstrated in 5.2.

Chapter 2

LITERATURE REVIEW

A survey of the pertinent literature reveals that apparently only one other worker, Warren (1991), has studied the particular aspect of flash calculations targeted in this research. A comprehensive review of the literature pertaining to the use of cubic equations of state in flash calculations was conducted in order to provide a reference point for the testing of the polynomial objective function.

This review is sub-divided into three sections: flash calculation algorithms; equilibrium ratios; and cubic equations of state. Particular emphasis is laid on the Peng-Robinson equation of state.

2.1 Vapor-Liquid Equilibrium Flash Calculations

This discussion will be confined to two-phase vapor-liquid equilibria. The work to date concentrates on developing robust algorithms with rapid convergence rates. Robustness implies the ability to continue the calculations after recovering from a spurious value of the vapor fraction computed in the neighborhood of the critical point or at the phase boundaries. Abhvani and Beaumont (1987) present an excellent review of EOS-based flash algorithms. They divide the papers into two categories according to solution method, those using some variant of the successive substitution method (SSM) or those employing a second-order Newton-type method.

The SSM technique is the traditional solution algorithm, but it exhibits a poor rate of convergence and has stability problems close to saturation points and in the critical region. Risnes et al. (1981), Michelsen (1982), and Mehra et al. (1983) made attempts at acceleration and stabilization of this method.

Similarly, many workers have proposed various forms of second-order Newton procedures to avoid the slow rate of convergence of the SSM, such as Boston and Britt (1978), Fussell and Yanosik (1978), Asselineau et al. (1979), Fussell (1979), Baker and Luks (1980), and Varotsis et al. (1981). Others advocate a combination of successive-substitution and Newton methods; the former is used to provide good initial values to the rapidly converging latter. Informative studies include Mott (1980, 1983), Mehra et al. (1982), Michelsen (1982), Nghiem et al. (1983), and Abhvani and Beaumont (1987).

Benmekki (1984) developed a general algorithm for flash calculations that can utilize any cubic equation of state and features a specified calculational path for computing the phase boundaries. This is an attempt to ensure that bubblepoint and dewpoint computations originate from within the two-phase region, thus guaranteeing meaningful values of the equilibrium ratios.

Warren (1991) made a radical departure from previous efforts at enhancing flash calculation algorithms when he formulated an explicit linear equation for the vapor fraction of a binary system. He successfully extended this to a quadratic equation for a ternary system and a cubic equation for a quaternary mixture. The success achieved by Warren and the possibility of the existence of a generalized polynomial expression for the vapor fraction in a two-phase, *N*-component fluid system motivated the current work.

2.2 Vapor-Liquid Equilibrium Ratios

The use of initial equilibrium ratios close to the final values for a multicomponent fluid is crucial to the rapid convergence of any flash calculation. Experimental values are preferred because the prediction of K_i for a particular fluid at various combinations of temperature, pressure and composition requires lengthy calculations. Therefore, predictive methods for K-values are a limiting factor in the speed and robustness of any flash calculation algorithm.

The expression "equilibrium constant" was coined by Souders et al. (1932) and was defined as the ratio of the vapor mole fraction to that of the liquid. The basis for most predictive methods had its genesis when Cox (1923) observed that the lines on a semilogarithmic plot of vapor pressure against temperature appeared to converge to a single pressure. Katz and Hachmuth (1937) demonstrated an analogous behavior for equilibrium constants; they converged to unity at a fluid mixture's critical pressure.

White and Brown (1942) attempted to develop a correlation for K-values based on this "convergence" pressure. Hanson and Brown (1945) used experimental data to correlate the convergence pressure (p_k) at one temperature as a function of the molal average boiling point of the equilibrium vapor and liquid. They also showed that the convergence pressure concept could be extended from binary to multicomponent systems.

Hadden (1948, 1953) produced nomographs for equilibrium constants of pure components as functions of temperature and pressure, and incorporated convergence and vapor pressures into nomographs for mixtures. He demonstrated that mixture convergence pressure is a function of the operating temperature and of the liquid-phase composition exclusive of the lightest component concentration. This composition dependence led Muskat (1949) to propose the use of the term "equilibrium ratio" in place of "equilibrium constant." Edmister (1949) presented a graph involving the ratio of differences between the convergence and critical pressures and the ratio of differences between the system and critical temperatures.

Winn (1952) developed nomographs based on Hadden's (1948) results that allow the determination of K-values at a convergence pressure of 5000 psia. For systems with $p_k \neq 5000$, he provides a translation to find the value of K_i at other "apparent" convergence pressures. The methods proposed by these three authors require charts and do not lend themselves to computer calculations.

Hoffmann et al. (1953) attempted to extend Cox's (1923) vapor pressure graph for the purpose of determining equilibrium ratios by plotting log Kp against the component characterization factor F, where

Kp = product of equilibrium ratio and pressure

$$F = b \left[\frac{1}{T_B} - \frac{1}{T} \right] \tag{2.1}$$

b = constant required to translate the vapor pressure curve for a hydrocarbon onto the straight line of the Cox chart

T_B = hydrocarbon boiling point

T = system temperature

Brinkman and Sicking (1960) presented an iterative method for finding the convergence pressure based on the slope, s_P, of the plot mentioned in Hoffmann et al. (1953). Then, the equilibrium ratio could be determined as

$$K = \frac{p_k}{p} e^{s_P F} \tag{2.2}$$

Standing (1979) observed that the composition dependence of the equilibrium ratio is negligible at pressures below 1000 psia. He proceeded to combine the work of Hoffmann et al. (1953) and Brinkman and Sicking (1960) to develop a correlation for K-values for the crude oils studied by Katz and Hachmuth (1937):

$$K = \frac{1}{p} 10^{(a + cF)} \tag{2.3}$$

where a and c are the intercept and slope (respectively) of log Kp vs. F plots of the abovementioned oils. Both a and c are expressed as functions of pressure. He also presented equations for the heavy fraction and the common reservoir gases N_2, CO_2 and H_2S (referred to as permanent gases).

Wilson (1969) published a K-value equation that currently enjoys widespread use in flash calculations:

$$K_i = \frac{e^B}{p_{Ri}} \tag{2.4}$$

where

$$B = 5.37(1 + \omega_i)(1 - \frac{1}{T_{Ri}}) \tag{2.5}$$

p_{Ri} = reduced pressure of the i-th component
T_{Ri} = reduced temperature of the i-th component
ω_i = Pitzer acentric factor of the i-th component

Wilson's equation fails to predict accurate equilibrium ratios for most fluids above pressures of 500 psia, as illustrated by Warren (1991). Whitson and Torp (1981) attempted to correct this problem by re-introducing the system convergence pressure to the Wilson equation:

$$K_i = \left[\frac{p_{ci}}{p_k} \right]^{A - 1} \frac{e^{AB}}{p_{Ri}} \tag{2.6}$$

where

$$A = 1 - \left[\frac{p - 14.7}{p_k - 14.7} \right]^{0.6} \tag{2.7}$$

p_{ci} = critical pressure of the i-th component

Risnes and Dalen (1984) took an approach based on the equation of state used in the flash calculations. Their basic idea was to assume the mixture or feed to be liquid and then evaporate up to one-half of the system to form a gas phase by use of the fugacities. The initial K-values then could be calculated from the resulting phases. This method is reported to perform well near the critical point and along the bubblepoint line but often fails along the dewpoint curve.

Reportedly, the most accurate K-value predictor is that proposed by Varotsis (1989). He used over 1000 experimental equilibrium ratios to construct an X-Y plot such that each reservoir fluid's position on the "map" is determined by its coordinates X and Y. These coordinates are described by a polynomial fitted to the apparent pressure mentioned in Winn (1952). He proposes an equation for the convergence pressure based on the mole fraction of methane and nitrogen in the fluid.

Each pure hydrocarbon component is represented on the map by its own set of coordinates (X_i, Y_i), which are calculated as functions of the component acentric factor. Specific values are given for the permanent gases and correlations based on molecular weight are specified for the heptane-plus fraction.

Finally, the straight line that joins the pressure and temperature coordinates (X, Y) of the fluid with the position of each component on the map (X_i, Y_i) intersects the K-value axis at a point that corresponds to the equilibrium value of the selected constituent. Varotsis (1989) presents tables for three different crude oils and gas condensates at varying temperatures and pressures that display K-values remarkably close to experimental values. They are an order-of-magnitude improvement over those predicted by the equations of Wilson (1969) or Whitson and Torp (1981).

The method of Varotsis (1989) was attempted in the current work. His K-value predictor was formulated using data for crude oils and gas condensates containing the $C_1 - C_6$ alkane series, the heptane-plus pseudocomponent and the permanent gases. It will not properly describe systems (such as the methane-ethane-propane ternary) containing fewer components than these "typical" reservoir fluids. For lack of a suitable replacement expression for p_k, Wilson's equation is used in the current work.

2.3 Cubic Equations of State (EOS)

The equation of state (EOS) is the heart of a modern flash calculation algorithm. Ideally, it should be able to accurately represent the thermodynamic properties of the fluid of interest over the complete range of operating pressures and temperatures. Since engineering applications rarely focus on an isolated chemical species, the EOS should incorporate mixing rules that allow it to extend its predictive capabilities to the behavior of multicomponent fluids. Its component-specific descriptive parameters should be readily calculable from well-known properties, such as critical temperature and pressure, molecular weight and acentric factor. Finally, the associated computations should not consume excessive computer time, especially if the equation of state is to be used for repetitive calculations.

The engineer is faced with the choice of using a complex EOS exhibiting a high degree of non-linearity and many adjustable parameters, or a cubic EOS which possesses an analytical solution and requires the estimation of two or three parameters. Mathias and Benson (1986) presented a comparison of average central-processing-unit (CPU) times required by three cubic EOS and by three complicated EOS to compute fugacity coefficients and enthalpy departures. They asserted that the time required for any of the candidate equations to calculate the density root (or compressibility factor root) is negligible compared to that involved in executing the various mixing rules. In fact, for systems containing more than about six components, the cubic EOS become more computationally burdensome than the complicated EOS simply because of the cross terms inherent to the cubic EOS mixing rules.

Engineers frequently use cubic EOS because they work well over the range of most industrial operating conditions and are easily programmed for solution on a computer. The two cubic EOS which have gained the widest acceptance are Soave's modifications of the Redlich-Kwong (1949) equation of state (SRKEOS) (Soave, 1972) and that presented by Peng and Robinson (1976b) (PREOS). The PREOS and suggested improvements are examined in this work for possible use in flash calculations because of the author's familiarity with this EOS.

2.3.1 Development of the Peng-Robinson EOS

Upon the success of the SRKEOS, Peng et al. (1975) undertook a further study to formulate a cubic equation of state with an improved capability to predict liquid densities and other fluid properties, particularly in the vicinity of the critical region.

This study resulted in a further modification of the attractive pressure term of the classical equation of state proposed by van der Waals (1873). The result was the EOS presented by Peng and Robinson (1976b):

$$p = \frac{RT}{v-b} - \frac{a(T)}{v(v+b) + b(v-b)} \tag{2.8}$$

The values of the parameters are obtained from

$$b = 0.07780 \frac{RT_c}{p_c} \tag{2.9}$$

$$a(T_c) = 0.45724 \frac{R^2 T_c^{\ 2}}{p_c} \tag{2.10}$$

$$a(T) = a(T_c) \cdot \eta(T_R, \omega) \tag{2.11}$$

$$\eta^{½} = 1 + \kappa\,(1 - T_R^{\ ½}) \tag{2.12}$$

$$\kappa = 0.3746 + 1.4850\omega - 0.1644\omega^2 + 0.01667\omega^3 \tag{2.13}$$

Equation (2.12) has the same form as that used by Soave (1972) but κ was obtained by fitting a larger range of vapor pressure data as a function of the reduced temperature and the acentric factor (Pitzer et al., 1955) of each component.

In order to use the equation for systems containing more than one component, the following mixing rules are presented:

$$a = \sum_i \sum_j x_i x_j a_{ij} \tag{2.14}$$

$$b = \sum_i x_i b_i \tag{2.15}$$

where

$$a_{ij} = (1 - \delta_{ij}) a_i^{\ ½} a_j^{\ ½} \tag{2.16}$$

Equations (2.14) and (2.15) are consequences of the mixing rule proposed by Kay (1936), while Equation (2.16) was developed by Zudkevitch and Joffe (1970). The experimentally determined binary interaction coefficient, δ_{ij}, characterizes the binary formed by component i and component j. The importance of δ_{ij} in accurately reproducing P-V-T data was discussed by Peng and Robinson (1976b) and by Robinson et al. (1985).

The PREOS can be written in the form of a cubic equation in the compressibility factor:

$$Z^3 - (1 - B)Z^2 + (A - 3B^2 - 2B)Z - (AB - B^2 - B^3) = 0 \tag{2.17}$$

where

$$A = \frac{ap}{R^2T^2} \tag{2.18}$$

$$B = \frac{bp}{RT} \tag{2.19}$$

$$Z = \frac{pv}{RT} \tag{2.20}$$

2.3.2 Selection of the Proper Root in Cubic EOS

Equation (2.17) yields one or three roots depending upon the number of phases in the system. The authors stated that, in the two-phase region, the largest root is for the compressibility factor of the vapor while the smallest positive root corresponds to that of the liquid.

Lawal (1987), however, asserted that this criterion was insufficient to select the proper root. He proved that, in the event of multiple real roots, the smallest of the positive roots *larger than or equal to B* must be chosen for the compressibility of the liquid.

Asselineau et al. (1979) compared the calculated volume to the pseudo-critical volume to assign the root to the proper phase, under specific conditions. Poling et al. (1981) examined the order of magnitude of the isothermal compressibility, $\beta = -(\partial v/\partial p)_T/v$, to ascertain the presence of the liquid or vapor phase. Gosset et al. (1986) offered two discriminants, one based on the Cardan criterion for the number of real roots for a cubic equation and a heuristic approach similar to that of Asselineau et al. (1979).

2.3.3 Modifications to the Peng-Robinson EOS

Numerous attempts have been made to correct for the deficiencies inherent in a cubic equation of state by introducing additional parameters into the PREOS. These changes improve some aspect of the EOS's performance (usually liquid density predictions) but at the cost of increased complexity and the requirement for tables or correlations to determine the additional parameter(s) for each fluid component. This review will touch on a limited number of these studies.

2.3.3.1 Volume Corrections

The modification of the SRKEOS proposed by Péneloux et al. (1982) also formed the basis for two other studies concerned with the PREOS. These authors suggested that the use of a "pseudo volume" defined by

$$\tilde{v} = v + \sum_i c_i x_i \tag{2.21}$$

could be used to effect a translation along the volume axis, leaving unchanged the predicted equilibrium conditions. They chose c so that correct saturated liquid densities were exactly reproduced at the reduced temperature $T_R = 0.7$. They rejected the acentric factor as a correlating parameter in favor of the Rackett compressibility factor, Z_{RA}, developed by Spencer and Danner (1972):

$$c = 0.40768 \frac{RT_c}{p_c} (0.29441 - Z_{RA}) \tag{2.22}$$

Their third parameter did improve predictions of saturated liquid densities.

Almost simultaneously, Jhaveri and Youngren (1988) and Mathias et al. (1989) presented three-parameter modifications of the PREOS based on the work of Péneloux et al. (1982). The first authors correlated the third parameter, c, with molecular weight. The second study retained the Péneloux-Rauzy-Fréze volume correction scheme but added a further term involving the bulk modulus to handle the critical region. The bulk modulus is dimensionless and is defined as:

$$\delta = -\frac{v^2}{RT} \left(\frac{\partial p}{\partial v} \right)_T \tag{2.23}$$

From an examination of the graphs accompanying both publications, the work of Mathias et al. (1989) seems to produce results closer to the experimental values for saturated volumes and densities.

2.3.3.2 Temperature Dependence

Xu and Sandler (1987a,b) postulated that the molar co-volume term, b, is not independent of temperature and they disputed the fitting of vapor pressures used by Peng and Robinson (1976b) to characterize the attractive constant, a. They correlated vapor pressure and volume data for 16 components at both subcritical and supercritical conditions and proposed to replace the numeric coefficients of a and b found in Equation (2.9) and Equation (2.10) with:

$$\psi_a{}^{\mathrm{I}} = \sum_{i=0}^{2} a_i{}^{\mathrm{I}} T_R^i \tag{2.24}$$

and

$$\psi_b{}^{\mathrm{I}} = \sum_{i=0}^{2} b_i{}^{\mathrm{I}} T_R^i \tag{2.25}$$

where i refers to the species and I denotes either subcritical or supercritical conditions.

Wu and Sandler (1989) generalized the temperature-dependent parameters of Xu and Sandler (1987a,b) by performing least-squares fits of ψ_a and ψ_b as functions of acentric factor and reduced temperature. They were able to accomplish this task only for the n-alkane series because of insufficient data. For their intended application of the work (petroleum reservoir simulation), they envisioned the use of the fluid-specific parameters for the permanent gases, water and light ends and the generalized parameters for the heavy pseudocomponents.

Stryjek and Vera (1986a,b,c,d) re-worked Equation (2.13) to obtain a better reproduction of vapor pressure data at low reduced temperatures:

$$\kappa_0 = 0.378893 + 1.4897153\omega - 0.17131848\omega^2 + 0.0196554\omega^3 \tag{2.26}$$

and modified Equation (2.12) by the introduction of one compound-characteristic adjustable parameter, κ_1:

$$\kappa = \kappa_0 + \kappa_1\,(1 + T_R{}^{1/2})(0.7 - T_R) \tag{2.27}$$

Stryjek and Vera (1986b) and Proust and Vera (1989) listed values of κ_1 for over 160 compounds of industrial interest. Stryjek and Vera (1986d) and Wilczek-Vera and Vera (1987) examined mixing rules of varying complexity for use with the Peng-Robinson-Stryjek-Vera (PRSV) EOS. For the current work, the PRSV EOS with the original PREOS mixing rules (as formulated by Stryjek and Vera, 1986b) is used and produces noticeably better results than did the PREOS.

2.3.4 References on Cubic EOS

Abbott (1979) and Martin (1979) presented comprehensive reviews of cubic equations of state available at that time, and Vidal (1983) and Vera et al. (1984) updated the topic. Huron and Vidal (1979) proposed composition-dependent mixing rules while Mathias and Copeman (1983) discussed mixing rules dependent on volume. Finally, Peng and Robinson (1976b), Peng and Robinson (1977), Robinson and Peng (1978), Robinson (1979) and Peng (1986) developed specific applications of their EOS.

Chapter 3

DEVELOPMENT OF THE POLYNOMIAL FUNCTION FOR SIMPLE SYSTEMS

This chapter discusses the work published by Rachford and Rice (1952) and Warren (1991) on performing flash calculations. It shows the development of the Rachford-Rice objective function [Equation (1.1)] and extends Warren's work as a precursor to developing a generalized, multicomponent equation for the vapor fraction.

3.1 The Rachford-Rice Flash Objective Function

We will briefly examine the derivation of the Rachford-Rice objective function that is universally used today in flash calculations. After plotting its behavior, it will become plain why it is so difficult to solve by iterative techniques such as the Newton-Raphson method.

3.1.1 The Material Balance Development

Flash calculations are used to determine the compositions and quantities of the vapor and liquid phases at equilibrium which result when an N-component fluid of a particular composition is subjected to a particular pressure and temperature. The composition of the feed stream, F, is denoted by Σz_i and it flashes into L moles of liquid with composition Σx_i, and V moles of vapor with composition Σy_i. The resulting material balance equations are:

$$F = L + V \tag{3.1}$$

$$Fz_i = Lx_i + Vy_i \tag{3.2}$$

As defined in Chapter 1, the equilibrium ratio is:

$$K_i = \frac{y_i}{x_i} \tag{3.3}$$

and, rearranging, one is left with the equation:

$$y_i = x_i K_i \tag{3.4}$$

Substituting Equation (3.4) into Equation (3.2) yields:

$$Fz_i = Vx_i K_i + Lx_i \tag{3.5}$$

Simplify by isolating the x_i term and dividing through by F:

$$z_i = x_i \left[\frac{VK_i}{F} + \frac{L}{F} \right] \tag{3.6}$$

Dividing Equation (3.1) through by F and solving for $\frac{L}{F}$ yields:

$$\frac{L}{F} = 1 - \frac{V}{F} \tag{3.7}$$

Substituting Equation (3.7) in Equation (3.6) and simplifying the equation results in:

$$x_i = \frac{z_i}{1 + \frac{V}{F} (K_i - 1)} \qquad (3.8)$$

Imposing the constraint of $\sum_{i=1}^{N} x_i = 1$ on Equation (3.8) leaves:

$$1 = \sum_{i=1}^{N} \frac{z_i}{1 + \frac{V}{F} (K_i - 1)} \qquad (3.9)$$

Rearranging:

$$\sum_{i=1}^{N} \left[\frac{z_i}{1 + \frac{V}{F} (K_i - 1)} \right] - 1 = 0 \qquad (3.10)$$

Recalling Equation (3.4), we can write:

$$y_i = \frac{z_i K_i}{1 + \frac{V}{F} (K_i - 1)} \qquad (3.11)$$

Imposing the constraint of $\sum_{i=1}^{N} y_i = 1$ on Equation (3.11) yields:

$$\sum_{i=1}^{N} \left[\frac{z_i K_i}{1 + \frac{V}{F} (K_i - 1)} \right] - 1 = 0 \qquad (3.12)$$

Combining Equation (3.10) and Equation (3.12) leaves:

$$\sum_{i=1}^{N} \frac{z_i\,(K_i - 1)}{1 + \frac{V}{F}\,(K_i - 1)} = 0 \tag{3.13}$$

Defining the vapor fraction, α, as:

$$\alpha = \frac{V}{F} \tag{3.14}$$

and substituting Equation (3.14) into Equation (3.13) yield the Rachford-Rice objective function:

$$\sum_{i=1}^{N} \frac{z_i\,(K_i - 1)}{1 + \alpha\,(K_i - 1)} = 0 \tag{3.15}$$

3.1.2 A Graphic Representation of the Rachford-Rice Objective Function

As the authors noted, their formulation of the objective function was prone to instability near the values of α that represented the phase boundaries, namely, 0 and 1. They showed that the slope of the function near these points may be quite steep. It is this feature that tends to throw derivative-based root-finding techniques out of the two-phase region, yielding spurious roots.

Figure 3.1 depicts the behavior of the objective function over a wide range of α for a binary system of 70% methane and 30% ethane (Bloomer et al., 1953). Figure 3.2 does the same for a ternary system consisting of 85% methane, 10% ethane and 5% propane (Parikh et al., 1984). Although values of the vapor fraction

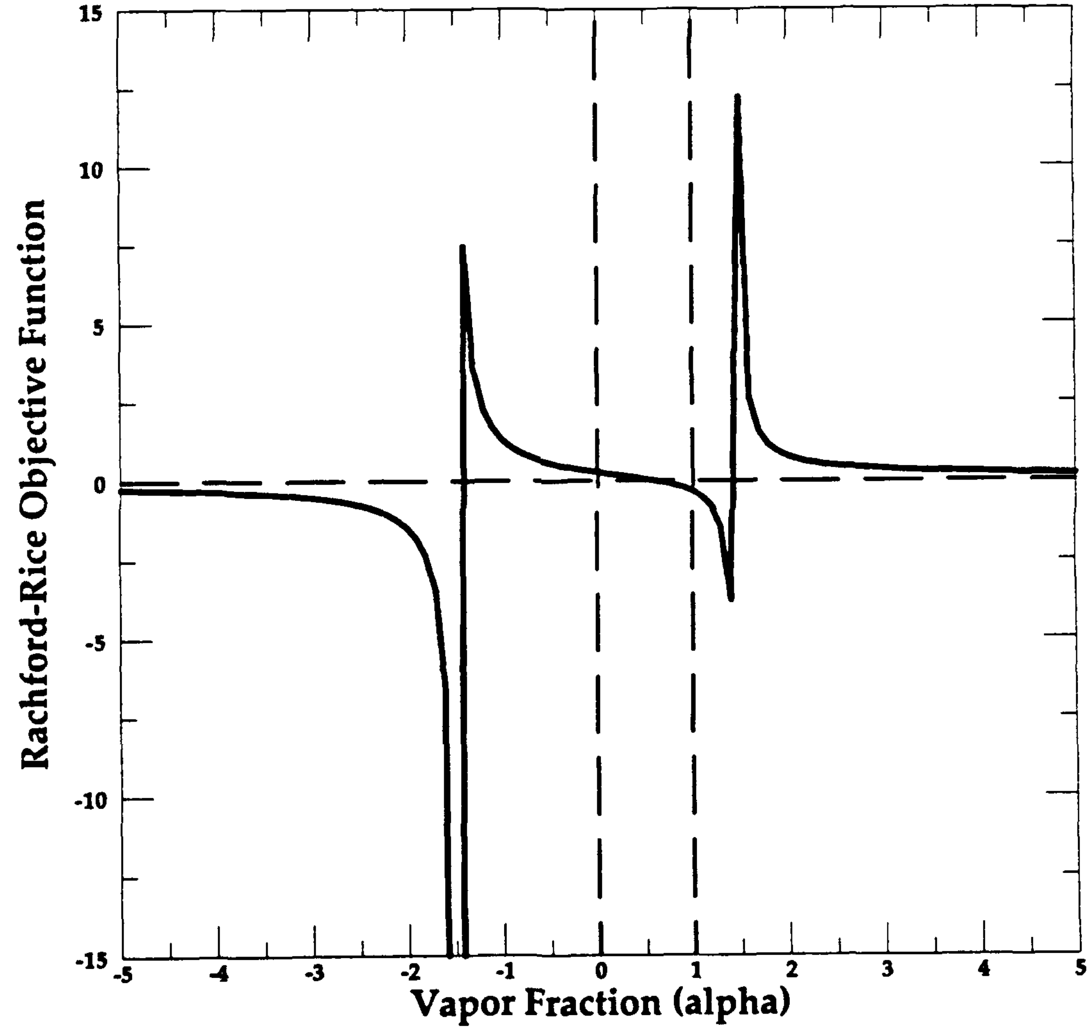


Figure 3.1: The Behavior of the Rachford-Rice Objective Function for a Methane-Ethane Binary System

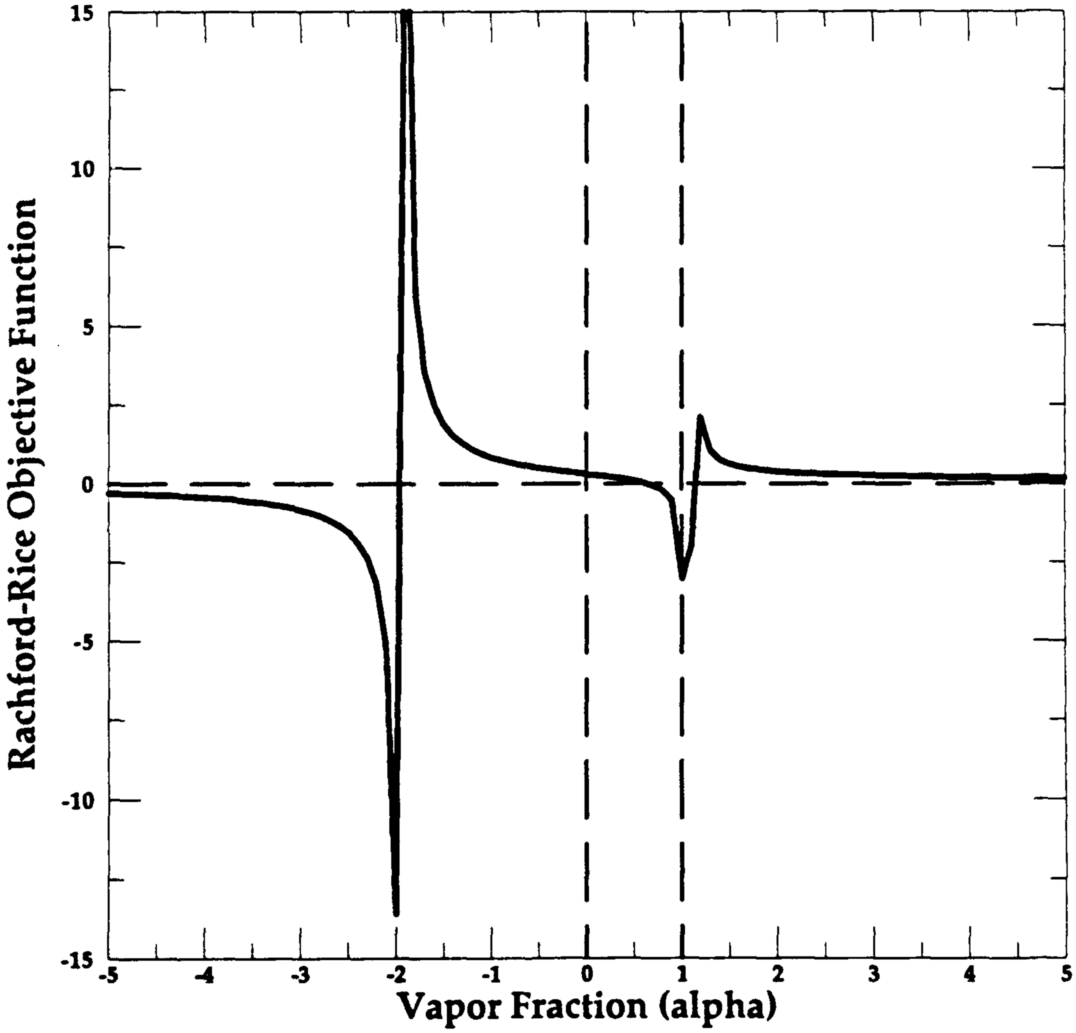


Figure 3.2: The Behavior of the Rachford-Rice Objective Function for a Methane-Ethane-Propane Ternary System

have no physical meaning outside the interval [0,1], these graphs serve to illustrate how ill-behaved the objective function is. Its slope is almost flat as it traverses the two-phase region and it is plagued by spiky singularities.

This work will attempt to develop a new expression for α, one that possesses reasonable slope over the desired interval and has no discontinuities near the phase boundaries.

3.2 Warren's Explicit Equations for the Vapor Fraction

Warren (1991) expanded the Rachford-Rice objective function into a polynomial in α for a binary, ternary and quaternary fluid system. He did this by setting N equal to 2, 3 or 4, respectively, and reducing the resulting equations to their simplest polynomial form by algebraic manipulations. To demonstrate the validity of his work, Warren also showed that the higher-order polynomials would reduce to those for smaller systems when the appropriate mole fractions and equilibrium constants were removed.

We will assume (as did Warren) that, under isobaric and isothermal conditions, the equilibrium constant does not change such that the quantity $(K_i - 1)$, which appears in the objective function, can be represented by a constant, C_i.

We will reproduce the entire process here for a binary system but will show only the final result for a ternary and quaternary system, since the algebra can be quite tedious and repetitive.

3.2.1 Binary System

Starting with the objective function as defined in Equation (3.15):

$$\sum_{i=1}^{N} \frac{z_i \, (K_i - 1)}{1 + \alpha \, (K_i - 1)} = 0 \tag{3.15}$$

and defining $C_i = K_i - 1$, Equation (3.15) can be rewritten as:

$$\sum_{i=1}^{N} \frac{z_i \, C_i}{1 + \alpha \, C_i} = 0 \tag{3.16}$$

For a two-component system, setting $N = 2$ in Equation (3.16) and expanding term-wise yields:

$$\frac{z_1 C_1}{1 + \alpha C_1} + \frac{z_2 C_2}{1 + \alpha C_2} = 0 \tag{3.17}$$

Moving the terms with the subscript "2" to the right-hand side of the equation:

$$\frac{z_1 C_1}{1 + \alpha C_1} = - \frac{z_2 C_2}{1 + \alpha C_2} \tag{3.18}$$

By multiplying each side by $(1 + \alpha \, C_1) \, (1 + \alpha \, C_2)$, one obtains:

$$(z_1 \, C_1) \, (1 + \alpha \, C_2) = - \, (z_2 \, C_2) \, (1 + \alpha \, C_1) \tag{3.19}$$

Expanding each side yields:

$$z_1 C_1 + \alpha \, z_1 C_1 C_2 = \qquad {}_{2} C_2 - \alpha \, z_2 C_1 C_2 \tag{3.20}$$

Move α terms to the left-hand side of the equation and all remaining terms to the right-hand side, then recall that, for a binary system, $z_1 + z_2 = 1$:

$$\alpha\,(C_1\,C_2) = -\,(z_1\,C_1 + z_2\,C_2) \tag{3.21}$$

Dividing both sides through by $C_1\,C_2$ and substituting $(K_i\,-\,1)$ yields the explicit form of the objective function for a binary system:

$$\alpha = -\left[\frac{z_1}{K_2 - 1} + \frac{z_2}{K_1 - 1}\right] \tag{3.22}$$

3.2.2 Ternary System

$$\alpha^2 + \alpha\left[\frac{\sum\limits_{i=1}^{3} z_i \left[\sum\limits_{j\neq i}^{3} C_j\right]}{\prod\limits_{j\neq i}^{3} C_j}\right] + \sum_{i=1}^{3} \frac{z_i}{\prod\limits_{j\neq i}^{3} C_j} = 0 \tag{3.23}$$

3.2.3 Quaternary System

$$\alpha^3 + \alpha^2\left[\sum_{i=1}^{4} \frac{(1 - z_i)}{C_i}\right] + \alpha\left[\frac{\sum\limits_{i=1}^{4} z_i \left[\sum\limits_{j\neq i}^{4} C_j\right]}{\prod\limits_{j\neq i}^{4} C_j}\right] + \sum_{i=1}^{4} \frac{z_i}{\prod\limits_{j\neq i}^{4} C_j} = 0 \tag{3.24}$$

3.3 Extension of Warren's Work to Larger Systems

Warren's method can be used to develop polynomial expressions for systems having five, six and seven components. It will be observed that the terms of the equation expand in a regular fashion, thereby suggesting the possibility of developing a recursive relationship dependent only on N, the number of components. Only the algebra for the five-component system will be presented, as those for six- and seven-component systems follow the same procedure.

3.3.1 Quinary System

We begin with the objective function:

$$\sum_{i=1}^{5} \frac{z_i\, C_i}{1 + \alpha\, C_i} = 0 \tag{3.16}$$

which, when expanded for five components, becomes:

$$\frac{z_1\, C_1}{1 + \alpha\, C_1} + \frac{z_2\, C_2}{1 + \alpha\, C_2} + \frac{z_3\, C_3}{1 + \alpha\, C_3} + \frac{z_4\, C_4}{1 + \alpha\, C_4} + \frac{z_5\, C_5}{1 + \alpha\, C_5} = 0 \tag{3.25}$$

Multiplying through by $\prod_{i=1}^{5} (1 + \alpha\, C_i)$ yields:

$$\begin{aligned}
& z_1\, C_1 (1 + \alpha\, C_2)\,(1 + \alpha\, C_3)\,(1 + \alpha\, C_4)\,(1 + \alpha\, C_5) + \\
& z_2\, C_2\,(1 + \alpha\, C_1)\,(1 + \alpha\, C_3)\,(1 + \alpha\, C_4)\,(1 + \alpha\, C_5) + \\
& z_3\, C_3\,(1 + \alpha\, C_1)\,(1 + \alpha\, C_2)\,(1 + \alpha\, C_4)\,(1 + \alpha\, C_5) + \\
& z_4\, C_4\,(1 + \alpha\, C_1)\,(1 + \alpha\, C_2)\,(1 + \alpha\, C_3)\,(1 + \alpha\, C_5) + \\
& z_5\, C_5\,(1 + \alpha\, C_1)\,(1 + \alpha\, C_2)\,(1 + \alpha\, C_3)\,(1 + \alpha\, C_4) \;=\; 0
\end{aligned} \tag{3.26}$$

Expanding each term:

$$\alpha^4\, z_1\, C_1\, C_2\, C_3\, C_4\, C_5 +$$

$$\begin{gathered}\alpha^3 z_1 C_1 (C_2 C_3 C_4 + C_2 C_3 C_5 + C_2 C_4 C_5 + C_3 C_4 C_5) + \\ \alpha^2 z_1 C_1 (C_2 C_3 + C_2 C_4 + C_2 C_5 + C_3 C_4 + C_3 C_5 + C_4 C_5) + \\ \alpha z_1 C_1 (C_2 + C_3 + C_4 + C_5) + z_1 C_1 +\end{gathered}$$

$$\begin{gathered}\alpha^4 z_2 C_1 C_2 C_3 C_4 C_5 + \\ \alpha^3 z_2 C_2 (C_1 C_3 C_4 + C_1 C_3 C_5 + C_1 C_4 C_5 + C_3 C_4 C_5) + \\ \alpha^2 z_2 C_2 (C_1 C_3 + C_1 C_4 + C_1 C_5 + C_3 C_4 + C_3 C_5 + C_4 C_5) + \\ \alpha z_2 C_2 (C_1 + C_3 + C_4 + C_5) + z_2 C_2 +\end{gathered}$$

$$\begin{gathered}\alpha^4 z_3 C_1 C_2 C_3 C_4 C_5 + \\ \alpha^3 z_3 C_3 (C_1 C_2 C_4 + C_1 C_2 C_5 + C_1 C_4 C_5 + C_2 C_4 C_5) + \\ \alpha^2 z_3 C_3 (C_1 C_2 + C_1 C_4 + C_1 C_5 + C_2 C_4 + C_2 C_5 + C_4 C_5) + \\ \alpha z_3 C_3 (C_1 + C_2 + C_4 + C_5) + z_3 C_3 +\end{gathered}$$

$$\begin{gathered}\alpha^4 z_4 C_1 C_2 C_3 C_4 C_5 + \\ \alpha^3 z_4 C_4 (C_1 C_2 C_3 + C_1 C_2 C_5 + C_2 C_3 C_5 + C_1 C_3 C_5) + \\ \alpha^2 z_4 C_4 (C_1 C_2 + C_1 C_3 + C_1 C_5 + C_2 C_3 + C_2 C_5 + C_3 C_5) + \\ \alpha z_4 C_4 (C_1 + C_2 + C_3 + C_5) + z_4 C_4 +\end{gathered}$$

$$\begin{gathered}\alpha^4 z_5 C_1 C_2 C_3 C_4 C_5 + \\ \alpha^3 z_5 C_5 (C_1 C_2 C_3 + C_1 C_2 C_4 + C_1 C_3 C_4 + C_2 C_3 C_4) + \\ \alpha^2 z_5 C_5 (C_1 C_2 + C_1 C_3 + C_1 C_4 + C_2 C_3 + C_2 C_4 + C_3 C_4) + \\ \alpha z_5 C_5 (C_1 + C_2 + C_3 + C_4) + z_5 C_5 = 0 \qquad (3.27)\end{gathered}$$

Dividing through by the term $\prod_{i=1}^{5} C_i$ and adding like terms yields:

$$\alpha^4 + \alpha^3 \left\{ z_1 \left[\frac{1}{C_2} + \frac{1}{C_3} + \frac{1}{C_4} + \frac{1}{C_5} \right] + z_2 \left[\frac{1}{C_1} + \frac{1}{C_3} + \frac{1}{C_4} + \frac{1}{C_5} \right] + \right.$$

$$z_3 \left[\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_4} + \frac{1}{C_5} \right] + z_4 \left[\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \frac{1}{C_5} \right] +$$

$$z_5\left[\frac{1}{C_1}+\frac{1}{C_2}+\frac{1}{C_3}+\frac{1}{C_4}\right]\Bigg\}+$$

$$\alpha^2\Bigg\{z_1\left[\frac{1}{C_2C_3}+\frac{1}{C_2C_4}+\frac{1}{C_2C_5}+\frac{1}{C_3C_4}+\frac{1}{C_3C_5}+\frac{1}{C_4C_5}\right]+$$

$$z_2\left[\frac{1}{C_1C_3}+\frac{1}{C_1C_4}+\frac{1}{C_1C_5}+\frac{1}{C_3C_4}+\frac{1}{C_3C_5}+\frac{1}{C_4C_5}\right]+$$

$$z_3\left[\frac{1}{C_1C_2}+\frac{1}{C_1C_4}+\frac{1}{C_1C_5}+\frac{1}{C_2C_4}+\frac{1}{C_2C_5}+\frac{1}{C_4C_5}\right]+$$

$$z_4\left[\frac{1}{C_1C_2}+\frac{1}{C_1C_3}+\frac{1}{C_1C_5}+\frac{1}{C_2C_3}+\frac{1}{C_2C_5}+\frac{1}{C_3C_5}\right]+$$

$$z_5\left[\frac{1}{C_1C_2}+\frac{1}{C_1C_3}+\frac{1}{C_1C_4}+\frac{1}{C_2C_3}+\frac{1}{C_2C_4}+\frac{1}{C_3C_4}\right]\Bigg\}+$$

$$\alpha\Bigg\{z_1\left[\frac{C_2+C_3+C_4+C_5}{C_2C_3C_4C_5}\right]+z_2\left[\frac{C_1+C_3+C_4+C_5}{C_1C_3C_4C_5}\right]+$$

$$z_3\left[\frac{C_1+C_2+C_4+C_5}{C_1C_2C_4C_5}\right]+z_4\left[\frac{C_1+C_2+C_3+C_5}{C_1C_2C_3C_5}\right]+$$

$$z_5\left[\frac{C_1+C_2+C_3+C_4}{C_1C_2C_3C_4}\right]\Bigg\}+\frac{z_1}{C_2C_3C_4C_5}+\frac{z_2}{C_1C_3C_4C_5}+$$

$$\frac{z_3}{C_1C_2C_4C_5}+\frac{z_4}{C_1C_2C_3C_5}+\frac{z_5}{C_1C_2C_3C_4}=0 \qquad (3.28)$$

To maintain similarity with the forms of the quaternary and ternary equations, we can separate the general term in the coefficient for α in Equation (3.28) into four fractions:

$$z_i\left[\frac{C_j+C_k+C_l+C_m}{C_j\,C_k\,C_l\,C_m}\right]=z_i\left[\frac{1}{C_k\,C_l\,C_m}+\frac{1}{C_j\,C_l\,C_m}+\frac{1}{C_j\,C_k\,C_m}+\frac{1}{C_j\,C_k\,C_l}\right] \qquad (3.29)$$

After multiplying each of the five fractions by z_i and collecting terms with common denominators, the following form appears:

$$\frac{z_i + z_j}{C_k \, C_l \, C_m} \tag{3.30}$$

We can invoke $\sum_{i=1}^{5} z_i = 1$ to construct fractions with similar subscripted terms in both numerator and denominator:

$$\frac{1 - z_k - z_l - z_m}{C_k \, C_l \, C_m} \tag{3.31}$$

This yields a final polynomial expression of the same general form as those of Warren (1991):

$$\begin{aligned}
&\alpha^4 + \alpha^3 \left\{ \frac{1-z_1}{C_1} + \frac{1-z_2}{C_2} + \frac{1-z_3}{C_3} + \frac{1-z_4}{C_4} + \frac{1-z_5}{C_5} \right\} + \\
&\alpha^2 \left\{ \frac{1-z_1-z_2}{C_1 C_2} + \frac{1-z_1-z_3}{C_1 C_3} + \frac{1-z_1-z_4}{C_1 C_4} + \frac{1-z_1-z_5}{C_1 C_5} + \frac{1-z_2-z_3}{C_2 C_3} + \right. \\
&\left. \frac{1-z_2-z_4}{C_2 C_4} + \frac{1-z_2-z_5}{C_2 C_5} + \frac{1-z_3-z_4}{C_3 C_4} + \frac{1-z_3-z_5}{C_3 C_5} + \frac{1-z_4-z_5}{C_4 C_5} \right\} + \\
&\alpha \left\{ \frac{1-z_1-z_2-z_3}{C_1 C_2 C_3} + \frac{1-z_1-z_2-z_4}{C_1 C_2 C_4} + \frac{1-z_1-z_2-z_5}{C_1 C_2 C_5} + \frac{1-z_1-z_3-z_4}{C_1 C_3 C_4} \right. \\
&+ \frac{1-z_1-z_3-z_5}{C_1 C_3 C_5} + \frac{1-z_1-z_4-z_5}{C_1 C_4 C_5} + \frac{1-z_2-z_3-z_4}{C_2 C_3 C_4} + \frac{1-z_2-z_3-z_5}{C_2 C_3 C_5} \\
&\left. + \frac{1-z_2-z_4-z_5}{C_2 C_4 C_5} + \frac{1-z_3-z_4-z_5}{C_3 C_4 C_5} \right\} + \frac{z_1}{C_2 C_3 C_4 C_5} + \frac{z_2}{C_1 C_3 C_4 C_5} + \\
&\frac{z_3}{C_1 C_2 C_4 C_5} + \frac{z_4}{C_1 C_2 C_3 C_5} + \frac{z_5}{C_1 C_2 C_3 C_4} = 0
\end{aligned} \tag{3.32}$$

This yields the polynomial expression for a quinary system:

$$\alpha^4 + \alpha^3 \left[\sum_{i=1}^{5} \frac{(1 - z_i)}{C_i} \right] + \alpha^2 \left[\sum_{i=1}^{4} \sum_{j=i+1}^{5} \frac{(1 - z_i - z_j)}{C_i\, C_j} \right] +$$

$$\alpha \left[\sum_{i=1}^{4} \sum_{j=i+1}^{5} \sum_{k=i+2}^{6} \frac{(1 - z_i - z_j - z_k)}{C_i\, C_j\, C_k} \right] + \sum_{i=1}^{5} \frac{z_i}{\prod\limits_{j \neq i}^{5} C_j} = 0 \qquad (3.33)$$

3.3.2 Reduction to Quaternary System

Before proceeding to develop the equations for six- and seven-component systems, we must ensure that the quinary equation will reduce to that of a quaternary system under the proper conditions. This is accomplished by setting z_5 equal to zero and K_5 equal to one (Warren, 1991).

When z_5 becomes zero, so must x_5 and y_5. This would seem to leave K_5 undefined:

$$\lim_{\substack{x_5 \to 0 \\ y_5 \to 0}} K_5 = \frac{0}{0} = \text{undefined} \qquad (3.34)$$

We can remove this difficulty by the application of l'Hospital's Rule. The expression becomes:

$$\lim_{\substack{x_5 \to 0 \\ y_5 \to 0}} K_5 = \frac{\dfrac{dy_5}{dy_5}}{\dfrac{dx_5}{dx_5}} = \frac{1}{1} = 1 \qquad (3.35)$$

Therefore, $C_5 = K_5 - 1 = 1 - 1 = 0$.

To avoid division by zero, multiply Equation (3.32) by $\prod_{i=1}^{5} C_i$:

$$
\begin{aligned}
&\alpha^4 C_1 C_2 C_3 C_4 C_5 + \\
&\alpha^3 \Big[(1 - z_1) C_2 C_3 C_4 C_5 + (1 - z_2) C_1 C_3 C_4 C_5 + (1 - z_3) C_1 C_2 C_4 C_5 + \\
&(1 - z_4) C_1 C_2 C_3 C_5 + (1 - z_5) C_1 C_2 C_3 C_4 \Big] + \\
&\alpha^2 \Big[(1 - z_1 - z_2) C_3 C_4 C_5 + (1 - z_1 - z_3) C_2 C_4 C_5 + (1 - z_1 - z_4) C_2 C_3 C_5 \\
&+ (1 - z_1 - z_5) C_2 C_3 C_4 + (1 - z_2 - z_3) C_1 C_4 C_5 + (1 - z_2 - z_4) C_1 C_3 C_5 + \\
&(1 - z_2 - z_5) C_1 C_3 C_4 + (1 - z_3 - z_4) C_1 C_2 C_5 + (1 - z_3 - z_5) C_1 C_2 C_4 + \\
&\qquad (1 - z_4 - z_5) C_1 C_2 C_3 \Big] + \\
&\alpha \Big[z_1 C_1 (C_2 + C_3 + C_4 + C_5) + z_2 C_2 (C_1 + C_3 + C_4 + C_5) + \\
&z_3 C_3 (C_1 + C_2 + C_4 + C_5) + z_4 C_4 (C_1 + C_2 + C_3 + C_5) + \\
&z_5 C_5 (C_1 + C_2 + C_3 + C_4) \Big] + z_1 C_1 + z_2 C_2 + z_3 C_3 + z_4 C_4 + z_5 C_5 = 0
\end{aligned}
\tag{3.36}
$$

Let z_5 and C_5 equal zero:

$$
\begin{aligned}
&\alpha^3 C_1 C_2 C_3 C_4 + \\
&\alpha^2 \Big[(1 - z_1) C_2 C_3 C_4 + (1 - z_2) C_1 C_3 C_4 + (1 - z_3) C_1 C_2 C_4 + \\
&\qquad (1 - z_4) C_1 C_2 C_3 \Big] + \\
&\alpha \Big[z_1 C_1 (C_2 + C_3 + C_4) + z_2 C_2 (C_1 + C_3 + C_4) + z_3 C_3 (C_1 + C_2 + C_4) + \\
&\qquad z_4 C_4 (C_1 + C_2 + C_3) + z_1 C_1 + z_2 C_2 + z_3 C_3 + z_4 C_4 = 0
\end{aligned}
\tag{3.37}
$$

This is identical to Equation (3.21) in Warren's work, which is the expanded form of the quaternary equation. We can now safely derive the expressions for six- and seven-component systems.

3.3.3 Senary System

$$\alpha^5 + \alpha^4 \left[\sum_{i=1}^{6} \frac{(1 - z_i)}{C_i} \right] + \alpha^3 \left[\sum_{i=1}^{5} \sum_{j=i+1}^{6} \frac{(1 - z_i - z_j)}{C_i\, C_j} \right] +$$

$$\alpha^2 \left[\sum_{i=1}^{4} \sum_{j=i+1}^{5} \sum_{k=i+2}^{6} \frac{(1 - z_i - z_j - z_k)}{C_i\, C_j\, C_k} \right] +$$

$$\alpha \left[\sum_{i=1}^{3} \sum_{j=i+1}^{4} \sum_{k=i+2}^{5} \sum_{l=i+3}^{6} \frac{(1 - z_i - z_j - z_k - z_l)}{C_i\, C_j\, C_k\, C_l} \right] + \sum_{i=1}^{6} \frac{z_i}{\prod_{j \neq i}^{6} C_j} = 0 \qquad (3.38)$$

3.3.4 Septenary System

$$\alpha^6 + \alpha^5 \left[\sum_{i=1}^{7} \frac{(1 - z_i)}{C_i} \right] + \alpha^4 \left[\sum_{i=1}^{6} \sum_{j=i+1}^{7} \frac{(1 - z_i - z_j)}{C_i\, C_j} \right] +$$

$$\alpha^3 \left[\sum_{i=1}^{5} \sum_{j=i+1}^{6} \sum_{k=i+2}^{7} \frac{(1 - z_i - z_j - z_k)}{C_i\, C_j\, C_k} \right] +$$

$$\alpha^2 \left[\sum_{i=1}^{4} \sum_{j=i+1}^{5} \sum_{k=i+2}^{6} \sum_{l=i+3}^{7} \frac{(1 - z_i - z_j - z_k - z_l)}{C_i\, C_j\, C_k\, C_l} \right] +$$

$$\alpha \left[\sum_{i=1}^{3} \sum_{j=i+1}^{4} \sum_{k=i+2}^{5} \sum_{l=i+3}^{6} \sum_{m=i+4}^{7} \frac{(1 - z_i - z_j - z_k - z_l - z_m)}{C_i\, C_j\, C_k\, C_l - C_m} \right]$$

$$+ \sum_{i=1}^{7} \frac{z_i}{\prod_{j \neq i}^{7} C_j} = 0 \qquad (3.39)$$

3.4 Formulation of a Generalized Equation for the Vapor Fraction

The objective function can be recast in the following form:

$$\sum_{i=1}^{N} z_i \; C_i \prod_{j \neq i}^{N} (1 + \alpha \; C_j) = 0, \text{ where } C_i, C_j = \text{constant} \tag{3.40}$$

Equation (3.40) is in the form of the generating function for the *elementary symmetric functions*, a_r:

$$\prod_{i=1}^{n} (1 + t \; x_i) = \sum_{r=0}^{n} t^r \; a_r(x_i) \tag{3.41}$$

According to Macdonald (1979), $a_0(x_i) = 1$ and $a_r(x_i) = 0$ for all $r > n$.

We can now express the objective function in terms of the r-th elementary symmetric function in C_i:

$$\sum_{i=1}^{N} z_i \; C_i \prod_{j \neq i}^{N} (1 + \alpha \; C_j) = \sum_{i=1}^{N} z_i \; C_i \sum_{r=0}^{N-1} \alpha^r \; a_r(C_1, \ldots, \hat{C}_i, \ldots, C_N) \; = \; 0 \tag{3.42}$$

where $\hat{C}_i$ indicates the exclusion of the i-th term from the operation.

Since α^r does not involve i, we can invert the order of the summations:

$$\sum_{r=0}^{N-1} \alpha^r \left\{ \sum_{i=1}^{N} z_i \; C_i \; a_r(C_1, \ldots, \hat{C}_i, \ldots, C_N) \right\} = 0 \tag{3.43}$$

A working definition of the elementary symmetric function a_r could be "taking permutations of the elements of a set r terms at a time." For example,

$$a_1(C_1, C_2, \ldots, C_N) \; = \; (C_1 + C_2 + \cdots + C_N) \tag{3.44}$$

$$a_2(C_1^{-1}, C_2^{-1}, \ldots, C_N^{-1}) = \left[\frac{1}{C_1 C_2} + \frac{1}{C_1 C_3} + \cdots + \frac{1}{C_{N-1}\, C_N} \right] \quad (3.45)$$

The condition $\hat{C}_i$ is equivalent to the $j \neq i$ condition imposed on the summation terms in the earlier versions of the α-polynomial and herein lies the computational awkwardness. We want to find an expression that allows the summation to proceed over *all* N components, which is an operation readily represented by a DO-loop in computer programming.

To eliminate $\hat{C}_i$, we must expand the symmetric function. In Chapter 4, we will tackle this problem after a discussion of symmetric functions.

Chapter 4

DEVELOPMENT OF THE GENERALIZED EQUATION

In this chapter, we shall present a brief introduction to the theory of symmetric functions to show why they provide such a powerful tool to express permutations. Then we will show the reasoning used in the search for a recursive expression for α in terms of N, C_i and z_i. Finally, we will present a generalized multicomponent equation for the vapor fraction, α, that is compact and readily programmed on a computer.

4.1 Introduction to Symmetric Functions

4.1.1 Notation and Definitions of Partitions

Any collection of v non-negative integers (excluding zero) whose sum is w is called a *v-partition* of w. The individual integers are referred to as *parts* of the partition and are conventionally written in descending order of magnitude.

David et al. (1966) state that if there are λ distinct parts, say $p_1, p_2, \ldots, p_\lambda$ with $p_1 > p_2 > p_3 > \cdots > p_\lambda \geq 1$ and if p_i is repeated π_i times, with $i = 1, 2, \ldots, \lambda$, then the partition is written $(p_1^{\pi_1} p_2^{\pi_2} \ldots p_\lambda^{\pi_\lambda})$. The *weight*, w, of the partition is written as

$$w = \sum_{i=1}^{\lambda} p_i \pi_i \tag{4.1}$$

and the number of parts, or *length*, is

$$\nu = \sum_{i=1}^{\lambda} \pi_i \tag{4.2}$$

Macdonald (1979) refers to π_i as the *multiplicity* of i in the partition. For example, the partition $(4^2 2 1^3)$ has weight 13, 6 parts and 3 distinct parts. In our notation,

$$p_1 = 4 \text{ and } \pi_1 = 2; \; p_2 = 2 \text{ and } \pi_2 = 1; \; p_3 = 1 \text{ and } \pi_3 = 3$$

4.1.2 Symmetric Functions

A *symmetric* function is one in which the individual parts can be interchanged without altering the value of the function, such as

$$\sum_{i=1}^{n} x_i = x_1 + x_2 + x_3 + \cdots + x_n \tag{4.3}$$

The number, n, of the quantities x does not affect the relationships between the various forms of the symmetric functions, but does appear in the final expressions. David et al. (1966) write

$$\sum_{i=1}^{n} x_i = (1), \quad \sum_{i=1}^{n} x_i^r = (r) \text{ and } \sum_{i \neq j}^{n} x_i^r x_j^s = (rs), \text{ for } r \neq s \tag{4.4}$$

This leads directly to the definitions of two special forms of symmetric functions. MacMahon (1920) defines the *unitary* or a-functions as

$$a_r = (\, 1^r \,) = \sum_{i_1 < \ldots < i_r}^{n} x_{i_1} \ldots x_{i_r} \, , \; r = 1, 2, \cdots \tag{4.5}$$

and the *power sums*, or s-functions, as

$$s_r = (r) = \sum_{i=1}^{n} x_i^r \, , \; r = 1, 2, \cdots \tag{4.6}$$

A special case of the a-function is the *augmented* unitary symmetric function, u_r (David et al., 1966):

$$u_r = [\ 1^r\ ] = r!(\ 1^r\ ) = r!a_r = \sum x_{i_1} \ldots x_{i_r}\ , \qquad (4.7)$$

summed over all ordered sets $i_1, \ldots, i_r$.

4.1.3 Recursive Expressions for Symmetric Functions

4.1.3.1 Interexpressibility Tables

Roe (1918) compiled comprehensive interexpressibility tables relating the various classes of symmetric functions to one another. These consist of a matrix of coefficients to be used in a polynomial which might yield, for example, $u_r = f(s_r)$. Of interest to this work is her relationship between the a-functions (often called *elementary* functions) and the s-functions; it is presented here in a form more clearly expressed by David et al. (1966):

$$a_r = \frac{1}{r!} u_r = \sum_{m=1}^{\lambda} \sum_{P} \frac{(-1)^{(r+m)}}{\pi_1! \ldots \pi_\lambda!} \frac{s_{p_1}^{\pi_1} \ldots s_{p_\lambda}^{\pi_\lambda}}{p_1^{\pi_1} \ldots p_\lambda^{\pi_\lambda}} \qquad (4.8)$$

David et al. (1966) also used this equation to construct interexpressibility tables describing polynomials in power-sum series (s) for a-functions up to and including weights of 12. For instance, a unitary symmetric function of weight 3 would be represented by the following polynomial from their Table 1.5.3:

$$a_3 = \frac{1}{3!} \left[(1)^3 - 3(2)(1) + 2(3) \right] \qquad (4.9)$$

which, in terms of s-functions, is:

$$a_3 = \frac{1}{3!}\left[s_1^3 - 3s_2s_1 + 2s_3 \right] \tag{4.10}$$

and, when written as power sums, becomes:

$$a_3 = \frac{1}{3!}\left[\left[\sum_{i=1}^{n} x_i \right]^3 - 3\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} x_i + 2\sum_{i=1}^{n} x_i^3 \right] \tag{4.11}$$

However, neither Equation (4.8) nor Equation (4.11) is conducive to solution by computer without a tremendous table look-up effort.

4.1.3.2 Determinant Form

Fortunately, David et al. (1966) present another relationship between a_r and s_r in determinant form:

$$a_r = \frac{1}{r!}u_r = \frac{1}{r!}\det \begin{bmatrix} s_1 & 1 & 0 & 0 & \cdots & 0 \\ s_2 & s_1 & 2 & 0 & \cdots & 0 \\ s_3 & s_2 & s_1 & 3 & \cdots & 0 \\ s_4 & s_3 & s_2 & s_1 & \cdots & 0 \\ . & . & . & . & \cdots & r-1 \\ s_r & s_{r-1} & s_{r-2} & s_{r-3} & \cdots & s_1 \end{bmatrix} = \frac{\det \mathbf{M}}{r!} \tag{4.12}$$

This provides a practical method of calculating a_r that is also readily programmable on a computer.

4.2 Search for a Recursive Expression for the Vapor Fraction

Armed with a working knowledge of symmetric functions, we can manipulate the α equation developed in Chapter 3:

$$\sum_{r=0}^{N-1} \alpha^{r} \left\{ \sum_{i=1}^{N} z_i \, C_i \, a_r(C_1, \ldots, \hat{C}_i, \ldots, C_N) \right\} = 0 \tag{3.43}$$

to eliminate the exclusion term $\hat{C}_i$ and expand the symmetric functions into a more recognizable form. We will examine the results for several values of r and use them to write a general expression for α as a function of N.

4.2.1 Case I: r = N−2

Equation (3.43) yields the following coefficient for α:

$$\alpha^{(N-2)} \left\{ \sum_{i=1}^{N} z_i \, C_i \, a_{N-2}(C_1, \ldots, \hat{C}_i, \ldots, C_N) \right\} \tag{4.13}$$

We can expand the symmetric function a_{N-2} as shown in Equation (4.14). Since the exclusion of C_i from the product on the RHS gives $(N-1)$ terms, we must subtract $\frac{1}{C_i}$ from the sum to yield $(N-2)$:

$$a_{N-2}(C_1, \ldots, \hat{C}_i, \ldots, C_N) = (C_1 \cdots \hat{C}_i \cdots C_N) \left[\frac{1}{C_1} + \frac{1}{C_2} + \cdots + \frac{1}{C_N} - \frac{1}{C_i} \right] \tag{4.14}$$

To eliminate $\hat{C}_i$, we can write the product on the RHS of Equation (4.14) as

$$(C_1 \cdots \hat{C}_i \cdots C_N) = \frac{(C_1 \cdots C_N)}{C_i} \tag{4.15}$$

This maneuver will allow the summation to proceed over all N components.

After substituting Equation (4.15) into Equation (4.14), we have:

$$a_{N-2}(C_1,\dots,\hat{C}_i,\dots,C_N) = \frac{(C_1 \cdots C_N)}{C_i}\left[\frac{1}{C_1}+\frac{1}{C_2}+\cdots+\frac{1}{C_N}-\frac{1}{C_i}\right] \tag{4.16}$$

Substituting Equation (4.16) into Equation (4.13), cancelling C_i, multiplying by z_i, and then summing over i gives:

$$\alpha^{(N-2)}\left\{(C_1 \cdots C_N)\left[\left[\sum_{i=1}^{N} z_i\right]\left[\frac{1}{C_1}+\frac{1}{C_2}+\cdots+\frac{1}{C_N}\right]-\sum_{i=1}^{N}\frac{z_i}{C_i}\right]\right\} \tag{4.17}$$

We recall that $\sum_{i=1}^{N} z_i = 1$ and recognize that $(C_1 \cdots C_N) = \prod_{k=1}^{N} C_k$. Noting the presence of an elementary symmetric function in $\left[\frac{1}{C_1}+\cdots+\frac{1}{C_N}\right]$, we can write Equation (4.17) as:

$$\alpha^{(N-2)}\left\{\left[\prod_{k=1}^{N} C_k\right]\left[a_1\left[\frac{1}{C_1},\dots,\frac{1}{C_N}\right]-\sum_{i=1}^{N}\frac{z_i}{C_i}\right]\right\} \tag{4.18}$$

4.2.2 Case II: r = N−3

Equation (3.43) now becomes:

$$\alpha^{(N-3)}\left\{\sum_{i=1}^{N} z_i\, C_i\, a_{N-3}\,(C_1,\dots,\hat{C}_i,\dots,C_N)\right\} \tag{4.19}$$

We can expand the symmetric function a_{N-3} as shown in Equation (4.20). We eliminate $\hat{C}_i$ in the same manner as in Equation (4.15) and remove $\frac{1}{C_i}$ in a similar fashion. But this also deletes the term $\frac{1}{C_i^2}$ which is necessary to cancel the corresponding term in the product. Therefore, we must compensate by adding $\frac{1}{C_i^2}$:

$$a_{N-3}(C_1,\dots,\hat{C}_i,\dots,C_N) = \frac{(C_1\cdots C_N)}{C_i}\left[\frac{1}{C_1C_2} + \frac{1}{C_1C_3} + \cdots \right.$$

$$\left. + \frac{1}{C_{N-1}C_N} - \frac{1}{C_i}\left[\frac{1}{C_1} + \frac{1}{C_2} + \cdots + \frac{1}{C_N}\right] + \frac{1}{C_i^2}\right] \tag{4.20}$$

Substituting Equation (4.20) into Equation (4.19) and making consolidations similar to the previous development yields:

$$\alpha^{(N-3)}\left\{\left[\prod_{k=1}^{N} C_k\right]\left[a_2\left[\frac{1}{C_1},\dots,\frac{1}{C_N}\right] - a_1\left[\frac{1}{C_1},\dots,\frac{1}{C_N}\right]\sum_{i=1}^{N}\frac{z_i}{C_i} + \sum_{i=1}^{N}\frac{z_i}{C_i^2}\right]\right\} \tag{4.21}$$

4.2.3 Case III: r = 0

We have saved consideration of this case for last because the properties of a_0 are not readily apparent. It would seem reasonable to interpret $a_0(C_1,\dots,\hat{C}_i,\dots,C_N)$ as meaning "taking permutations of the elements of a set *zero* terms at a time." However, when $r = 0$, $\alpha^r \rightarrow 1$ and we know from previous developments that our α-polynomial does have a constant term. Therefore, $a_0(C_1,\dots,\hat{C}_i,\dots,C_N)$ must equal *one*, after Macdonald (1979). So, for $r = 0$, Equation (3.44) becomes:

$$\left\{\sum_{i=1}^{N} z_i C_i\right\} \tag{4.24}$$

We show Equation (4.18) and Equation (4.21) again to look for patterns that may assist us in writing the expression for the $(N-p)$-th term:

$$\alpha^{(N-2)}\left\{\left[\prod_{k=1}^{N} C_k\right]\left[a_1\left[\frac{1}{C_1},\ldots,\frac{1}{C_N}\right]-\sum_{i=1}^{N}\frac{z_i}{C_i}\right]\right\} \tag{4.18}$$

$$\alpha^{(N-3)}\left\{\left[\prod_{k=1}^{N} C_k\right]\left[a_2\left[\frac{1}{C_1},\ldots,\frac{1}{C_N}\right]-a_1\left[\frac{1}{C_1},\ldots,\frac{1}{C_N}\right]\sum_{i=1}^{N}\frac{z_i}{C_i}+\sum_{i=1}^{N}\frac{z_i}{C_i^2}\right]\right\} \tag{4.21}$$

4.2.4 The General Case: r = (N−p), p = 1, 2, ..., N

After substituting for r, Equation (3.43) becomes:

$$\sum_{p=1}^{N}\alpha^{(N-p)}\left\{\sum_{i=1}^{N} z_i\; C_i\; a_{N-p}(C_1,\ldots,\hat{C}_i,\ldots,C_N)\right\} = 0 \tag{4.23}$$

By continuing the expansion of this equation in the same fashion as in the first two cases, we note a descending order of the symmetric function and an ascending exponent of C_i with each additional term. This leads to a general expression:

$$\begin{aligned}&\sum_{p=1}^{N}\alpha^{(N-p)}\left\{\sum_{i=1}^{N} z_i\;(C_1\cdots C_N)\Big[a_{p-1}(C_1^{-1},\ldots,C_N^{-1})\;-\right.\\ &C_i^{-1}a_{p-2}(C_1^{-1},\ldots,C_N^{-1})+C_i^{-2}a_{p-3}(C_1^{-1},\ldots,C_N^{-1})-C_i^{-3}a_{p-4}(C_1^{-1},\ldots,C_N^{-1})\\ &\left.\qquad+\;\cdots\;\pm C_i^{-(p-2)}a_1(C_1^{-1},\ldots,C_N^{-1})\;\pm C_i^{-(p-1)}\Big]\right\} = 0\end{aligned} \tag{4.24}$$

Multiply by z_i , sum i from 1 to N and recall that $(C_1 \cdots C_N) = \prod_{k=1}^{N} C_k$:

$$\sum_{p=1}^{N} \alpha^{(N-p)} \left\{ \left[\prod_{k=1}^{N} C_k \right] \left[a_{p-1} - a_{p-2} \sum_{i=1}^{N} \frac{z_i}{C_i} + a_{p-3} \sum_{i=1}^{N} \frac{z_i}{C_i^2} - \cdots + \right. \right.$$

$$\left. \left. (-1)^{(p-1)} \sum_{i=1}^{N} \frac{z_i}{C_i^{(p-1)}} \right] \right\} = 0 \qquad (4.25)$$

Since $\prod_{k=1}^{N} C_k$ does not involve p, we can move this term outside the summation sign and then divide it out as a factor common to all powers of α. By examining the relationship between p, the subscripts of a and the superscripts of C_i , we can collapse Equation (4.25) into a more compact form:

$$\sum_{p=1}^{N} \alpha^{(N-p)} \left\{ \sum_{j=1}^{p} \left[(-1)^{j+1} (a_{p-j}) \sum_{i=1}^{N} \frac{z_i}{C_i^{j-1}} \right] \right\} = 0 \qquad (4.26)$$

$$\text{where } a_{p-j} = a_{p-j}(C_1^{-1}, \ldots, C_N^{-1}) \qquad (4.27)$$

$$a_0 = 1 \qquad (4.28)$$

$$C_i = (K_i)_{T,P} - 1 \qquad (4.29)$$

4.3 A Generalized Multicomponent Equation for the Vapor Fraction

Using the determinant expression for the elementary symmetric functions that was presented in **4.1.3.2**, Equation (4.27) becomes:

$$a_{p-j} = a_{p-j}(\, C_1^{-1}, \ldots, C_N^{-1} \,) = \frac{\det \mathbf{M}}{(p-j)!} \tag{4.30}$$

The matrix **M** has dimensions $(p-j) \times (p-j)$ and elements given by:

$$[m_{kl}] = \begin{cases} s_{k-l+1} & \text{if } l \leq k \\ k & \text{if } l = k+1 \\ 0 & \text{if } l > k+1 \end{cases} \tag{4.31}$$

The s elements are given by $s_\lambda = \sum_{i=1}^{N} \left[\frac{1}{C_i}\right]^{\lambda}, \; \lambda = 1, 2, \ldots, (p-j)$ (4.32)

Chapter 5

VALIDATION OF THE GENERALIZED EQUATION

The first test of validity for Equation (4.26) requires that it be equivalent to the form of the objective function presented in Equation (3.40). Second, it must generate the same coefficients for the α polynomial that were produced through the expansion of the objective function in Equation (3.25) through Equation (3.32). Third, the equation must predict the proper vapor fraction for a fluid undergoing an isothermal, isobaric flash process.

The first test is supplied by a mathematical proof in Appendix A. The second test can be accomplished by comparing the coefficients produced by Equation (4.26) with those of Equation (3.33). Since this equation has already been shown to reduce to that for a quaternary system under the proper constraints on z_5 and K_5, then, by induction, we can state that the polynomial produced by Equation (4.26) will do the same and therefore should be valid for any number of components.

The third test will be satisfied by comparing the equilibrium ratios generated by Equation (4.26) with experimental values determined for several multicomponent hydrocarbon fluids.

5.1 The Generalized α Equation for a Quinary System

For a five-component system, Equation (4.26) becomes:

$$\sum_{p=1}^{5} \alpha^{(5-p)} \left\{ \sum_{j=1}^{p} \left[(-1)^{j+1} a_{p-j} \sum_{i=1}^{5} \frac{z_i}{C_i^{\,j-1}} \right] \right\} = 0 \qquad (5.1)$$

which will yield a quartic polynomial in α:

$$\mu_4\alpha^4 + \mu_3\alpha^3 + \mu_2\alpha^2 + \mu_1\alpha + \mu_0 = 0 \qquad (5.2)$$

5.1.1 Coefficient μ_4 (p = 1)

$$\mu_4 = a_0(C_i^{-1}) \left[z_1 + z_2 + z_3 + z_4 + z_5 \right] \qquad (5.3)$$

We have already said that $a_0(C_i^{-1})$ is defined as one and the sum of the mole fractions also equals one, so Equation (5.3) yields:

$$\mu_4 = 1 \qquad (5.4)$$

5.1.2 Coefficient μ_3 (p = 2)

$$\mu_3 = a_1(C_i^{-1}) \sum_{i=1}^{5} z_i - a_0(C_i^{-1}) \sum_{i=1}^{5} \frac{z_i}{C_i} \qquad (5.5)$$

$$\mu_3 = \left[\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \frac{1}{C_4} + \frac{1}{C_5} \right](1) -$$

$$(1)\left[\frac{z_1}{C_1} + \frac{z_2}{C_2} + \frac{z_3}{C_3} + \frac{z_4}{C_4} + \frac{z_5}{C_5} \right] \qquad (5.6)$$

$$\mu_3 = \frac{1 - z_1}{C_1} + \frac{1 - z_2}{C_2} + \frac{1 - z_3}{C_3} + \frac{1 - z_4}{C_4} + \frac{1 - z_5}{C_5} \qquad (5.7)$$

5.1.3 Coefficient μ_2 (p = 3)

$$\mu_2 = a_2(C_i^{-1}) \sum_{i=1}^{5} z_i - a_1(C_i^{-1}) \sum_{i=1}^{5} \frac{z_i}{C_i} + a_0(C_i^{-1}) \sum_{i=1}^{5} \frac{z_i}{C_i^2} \tag{5.8}$$

$$\begin{aligned}\mu_2 = {} & \left[\frac{1}{C_1C_2} + \frac{1}{C_1C_3} + \frac{1}{C_1C_4} + \frac{1}{C_1C_5} + \frac{1}{C_2C_3} + \frac{1}{C_2C_4} + \frac{1}{C_2C_5} + \frac{1}{C_3C_4} \right. \\ & \left. + \frac{1}{C_3C_5} + \frac{1}{C_4C_5} \right](1) - \left[\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \frac{1}{C_4} + \frac{1}{C_5} \right] \left[\frac{z_1}{C_1} + \frac{z_2}{C_2} + \frac{z_3}{C_3} \right. \\ & \left. + \frac{z_4}{C_4} + \frac{z_5}{C_5} \right] + (1)\left[\frac{z_1}{C_1^2} + \frac{z_2}{C_2^2} + \frac{z_3}{C_3^2} + \frac{z_4}{C_4^2} + \frac{z_5}{C_5^2} \right]\end{aligned} \tag{5.9}$$

$$\begin{aligned}\mu_2 = {} & \frac{1}{C_1C_2} + \frac{1}{C_1C_3} + \frac{1}{C_1C_4} + \frac{1}{C_1C_5} + \frac{1}{C_2C_3} + \frac{1}{C_2C_4} + \frac{1}{C_2C_5} + \frac{1}{C_3C_4} + \\ & \frac{1}{C_3C_5} + \frac{1}{C_4C_5} - \frac{z_1}{C_1^2} - \frac{z_2}{C_2^2} - \frac{z_3}{C_3^2} - \frac{z_4}{C_4^2} - \frac{z_5}{C_5^2} - \frac{z_1 + z_2}{C_1C_2} - \frac{z_1 + z_3}{C_1C_3} - \\ & \frac{z_1 + z_4}{C_1C_4} - \frac{z_1 + z_5}{C_1C_5} - \frac{z_2 + z_3}{C_2C_3} - \frac{z_2 + z_4}{C_2C_4} - \frac{z_2 + z_5}{C_2C_5} - \frac{z_3 + z_4}{C_3C_4} - \frac{z_3 + z_5}{C_3C_5} - \\ & \frac{z_4 + z_5}{C_4C_5} + \frac{z_1}{C_1^2} + \frac{z_2}{C_2^2} + \frac{z_3}{C_3^2} + \frac{z_4}{C_4^2} + \frac{z_5}{C_5^2}\end{aligned} \tag{5.10}$$

$$\begin{aligned}\mu_2 = {} & \frac{1 - z_1 - z_2}{C_1C_2} + \frac{1 - z_1 - z_3}{C_1C_3} + \frac{1 - z_1 - z_4}{C_1C_4} + \frac{1 - z_1 - z_5}{C_1C_5} + \frac{1 - z_2 - z_3}{C_2C_3} + \\ & \frac{1 - z_2 - z_4}{C_2C_4} + \frac{1 - z_2 - z_5}{C_2C_5} + \frac{1 - z_3 - z_4}{C_3C_4} + \frac{1 - z_3 - z_5}{C_3C_5} + \frac{1 - z_4 - z_5}{C_4C_5}\end{aligned} \tag{5.11}$$

5.1.4 Coefficient μ_1 (p = 4)

$$\mu_1 = a_3(C_i^{-1})\sum_{i=1}^{5} z_i - a_2(C_i^{-1})\sum_{i=1}^{5}\frac{z_i}{C_i} + a_1(C_i^{-1})\sum_{i=1}^{5}\frac{z_i}{C_i^2} - a_0(C_i^{-1})\sum_{i=1}^{5}\frac{z_i}{C_i^3} \tag{5.12}$$

$$\begin{aligned}\mu_1 = \Bigg[& \frac{1}{C_1C_2C_3} + \frac{1}{C_1C_2C_4} + \frac{1}{C_1C_2C_5} + \frac{1}{C_1C_3C_4} + \frac{1}{C_1C_3C_5} + \\ & \frac{1}{C_1C_4C_5} + \frac{1}{C_2C_3C_4} + \frac{1}{C_2C_3C_5} + \frac{1}{C_2C_4C_5} + \frac{1}{C_3C_4C_5}\Bigg](1) - \\ & \left[\frac{1}{C_1C_2} + \frac{1}{C_1C_3} + \frac{1}{C_1C_4} + \frac{1}{C_1C_5} + \frac{1}{C_2C_3} + \frac{1}{C_2C_4} + \frac{1}{C_2C_5} + \frac{1}{C_3C_4} + \right. \\ & \left.\frac{1}{C_3C_5} + \frac{1}{C_4C_5}\right]\left[\frac{z_1}{C_1} + \frac{z_2}{C_2} + \frac{z_3}{C_3} + \frac{z_4}{C_4} + \frac{z_5}{C_5}\right] + \\ & \left[\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \frac{1}{C_4} + \frac{1}{C_5}\right]\left[\frac{z_1}{C_1^2} + \frac{z_2}{C_2^2} + \frac{z_3}{C_3^2} + \frac{z_4}{C_4^2} + \frac{z_5}{C_5^2}\right] - \\ & (1)\left[\frac{z_1}{C_1^3} + \frac{z_2}{C_2^3} + \frac{z_3}{C_3^3} + \frac{z_4}{C_4^3} + \frac{z_5}{C_5^3}\right]\end{aligned} \tag{5.13}$$

It is evident that the $\dfrac{z_i}{C_i^2C_j}$ terms in the second part of Equation (5.13) will cancel those in the third part, while the $\dfrac{z_i}{C_i^3}$ terms in the third part will negate the entire fourth part of the equation. The first and second parts yield:

$$\mu_1 = \frac{1 - z_1 - z_2 - z_3}{C_1C_2C_3} + \frac{1 - z_1 - z_2 - z_4}{C_1C_2C_4} + \frac{1 - z_1 - z_2 - z_5}{C_1C_2C_5} + \frac{1 - z_1 - z_3 - z_4}{C_1C_3C_4}$$

$$+\frac{1-z_1-z_3-z_5}{C_1C_3C_5}+\frac{1-z_1-z_4-z_5}{C_1C_4C_5}+\frac{1-z_2-z_3-z_4}{C_2C_3C_4}+\frac{1-z_2-z_3-z_5}{C_2C_3C_5}+$$

$$\frac{1-z_2-z_4-z_5}{C_2C_4C_5}+\frac{1-z_3-z_4-z_5}{C_3C_4C_5} \tag{5.14}$$

5.1.5 Coefficient μ_0 (p = 5)

$$\mu_0 = a_4(C_i^{-1})\sum_{i=1}^{5} z_i - a_3(C_i^{-1})\sum_{i=1}^{5}\frac{z_i}{C_i} + a_2(C_i^{-1})\sum_{i=1}^{5}\frac{z_i}{C_i^2} -$$

$$a_1(C_i^{-1})\sum_{i=1}^{5}\frac{z_i}{C_i^3} + a_0(C_i^{-1})\sum_{i=1}^{5}\frac{z_i}{C_i^4} \tag{5.15}$$

The analogous cancellations of the higher-order $\frac{1}{C_i}$ terms will occur, leaving a sum of five terms having the form

$$\frac{1-z_i-z_j-z_k-z_l}{C_i\ C_j\ C_k\ C_l} \tag{5.16}$$

Since the mole fractions must sum to one, we can replace the numerator of Equation (5.16) with the mole fraction of the remaining component, z_m , to yield:

$$\mu_0 = \frac{z_1}{C_2C_3C_4C_5}+\frac{z_2}{C_1C_3C_4C_5}+\frac{z_3}{C_1C_2C_4C_5}+$$

$$\frac{z_4}{C_1C_2C_3C_5}+\frac{z_5}{C_1C_2C_3C_4} \tag{5.17}$$

A term-by-term comparison with Equation (3.32) shows that the generalized α polynomial [Equation (4.26)] produces identical results.

5.2 Reproduction of Experimental Vapor-Liquid Equilibrium Data

5.2.1 Flash Calculation Package

The flash calculation package used in this work incorporates the K-value equation of Wilson (1969) and the modified PREOS proposed by Stryjek and Vera (1986a). An attempt was made to use the K-value prediction of Varotsis (1989) but, as noted in Chapter 2, it was developed to characterize a broad-spectrum petroleum reservoir condensate or crude oil. It experiences difficulty handling an arbitrary hydrocarbon mixture, such as the artificial systems for which equilibrium data is available to validate this work.

The volume correction of Mathias et al. (1989) and the complementary calculation of Schick and Prausnitz (1968) for mixture pseudo-critical volume are incorporated into the PRSV EOS but since it is only required to generate compressibility factors and fugacities, the modifications have no noticeable effect on the computations. The PRSV EOS shows marked improvement over the PREOS when used to duplicate bubblepoint and dewpoint studies performed by Warren (1991).

The binary interaction coefficients used in the PRSV EOS are taken from Knapp et al. (1982) and Walas (1985). Physical property data and equation parameters for the chemical components are extracted from Stryjek and Vera (1986b,c), Kumar (1987) and Proust and Vera (1989).

The computation of the determinant used to generate the elementary symmetric functions is accomplished with a modified Gaussian elimination routine. The first elementary symmetric function, a_1, is defined by a [1×1] matrix, whose determinant constitutes the element itself. By definition, a_0 is set equal to one.

The polynomial is evaluated at the bubblepoint line ($\alpha = 0$) and an interval-halving technique is used to march across the two-phase region until the value of the polynomial changes sign, indicating the vicinity of the root. Then a Newton-Raphson iterative search is conducted to converge to the exact value of α.

5.2.2 Binary System

The fugacity-based flash algorithm is used to replicate the equilibrium ratios determined by Bloomer et al. (1953) for a methane-ethane system at a temperature of −60 °F over a pressure range of 100-900 psia. Comparisons of calculated and empirical values of K_{CH_4} and $K_{C_2H_6}$ appear in Figure 5.1 and 5.2, respectively. The results lie within the margin of error attributable to the PRSV EOS.

5.2.3 Septenary System

Standing (1977) provides a sample flash calculation for a seven-component hydrocarbon system reported by Dodson and Standing (1941), complete with values for experimental K_i and the vapor fraction. This sort of data allows the calculation of α solely on the basis of computing the coefficients of the α-polynomial and determining the applicable root, with no recourse to the equation of state. Once the interval-halving search provides an initial estimate of the root, the Newton-Raphson technique converges in three iterations to a value of α identical to that calculated by Standing.

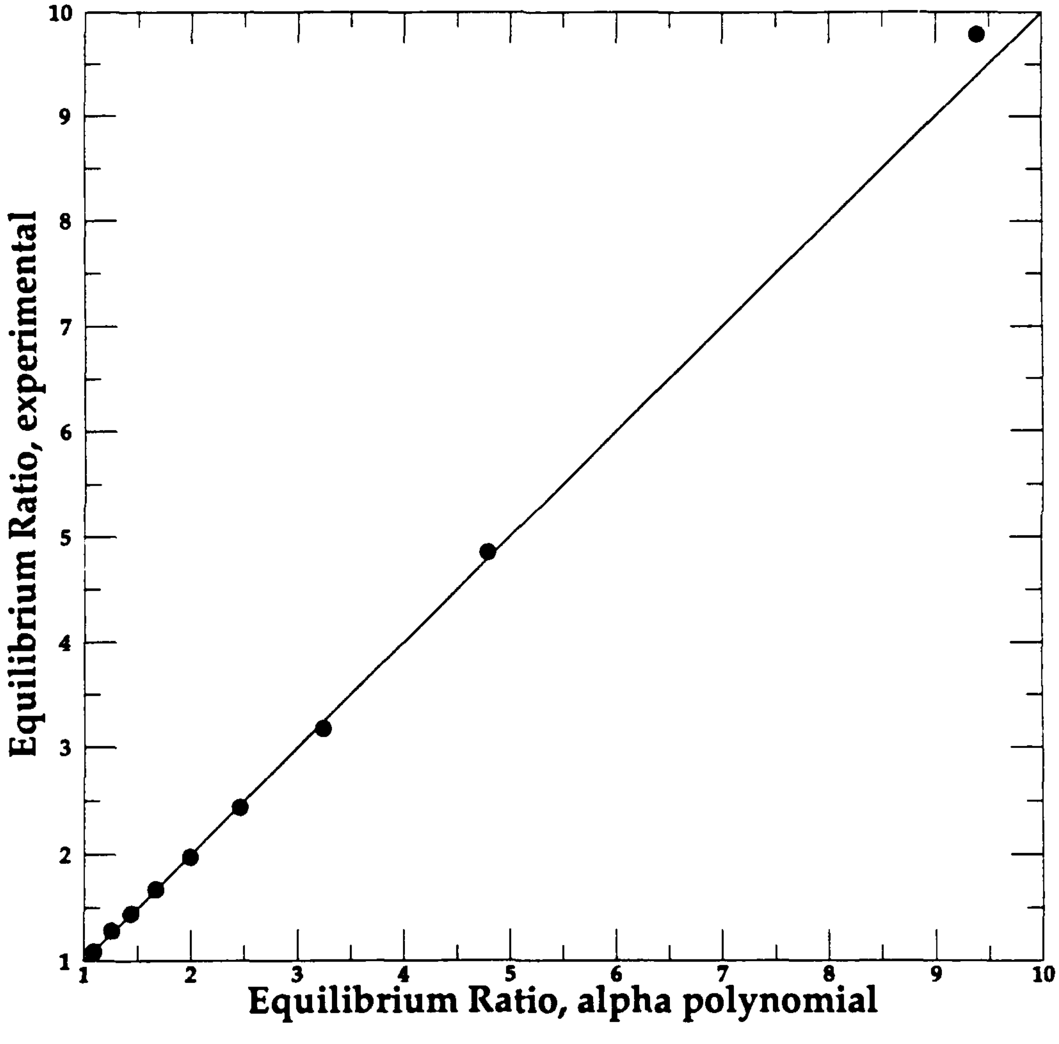


Figure 5.1: A Comparison of Predicted and Experimental Equilibrium Ratios for Methane

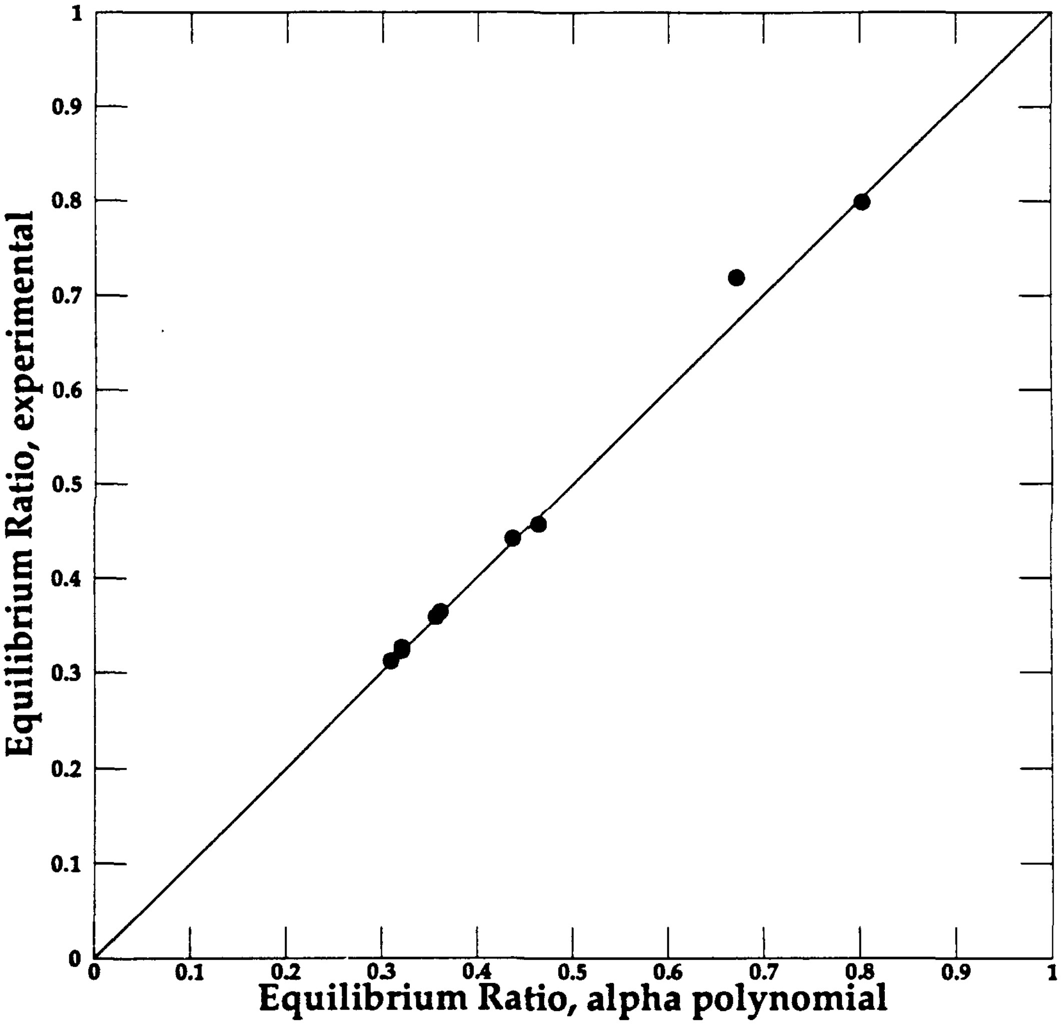


Figure 5.2: A Comparison of Predicted and Experimental Equilibrium Ratios for Ethane

5.2.4 Predicting Roots with the Fourier-Budan Theorem

A useful theorem for predicting the number of roots of a polynomial that can occur on a particular interval is that of Fourier and Budan (Barbeau, 1989). Suppose $p(t)$ is a polynomial over the field of real numbers, **R**, and that u and v are real numbers with $u < v$ and $p(u) \cdot p(v) \neq 0$. The number of zeros between u and v cannot be greater than $A - B$, where A is the number of changes of sign in the sequence $\{ p(u), p'(u), p''(u), \ldots, p^{(n)}(u) \}$ and B is the number of changes of sign in the sequence $\{ p(v), p'(v), p''(v), \ldots, p^{(n)}(v) \}$. If this number differs from $A - B$, it must do so by an even amount.

An interesting aspect of the polynomial expression for the vapor fraction is its capability to mathematically confirm the existence of a unique value within the two-phase region for a particular set of feed conditions. This is equivalent to stating that the polynomial has only one zero on the interval $0 \leq \alpha \leq 1$. From the physics of the problem, we know this to be true but, by the use of the Fourier-Budan theorem, we can also prove it rigorously.

Let us test this theorem on the septenary system of Standing (1977) utilized in **5.2.3**; this is represented by a sixth-order polynomial:

$$\mu_6\alpha^6 + \mu_5\alpha^5 + \mu_4\alpha^4 + \mu_3\alpha^3 + \mu_2\alpha^2 + \mu_1\alpha + \mu_0 = 0 \tag{5.18}$$

where

$\mu_0 = -9.58519$ $\quad \mu_3 = 87.24949$

$\mu_1 = 65.90501$ $\quad \mu_4 = -21.71701$

$\mu_2 = -120.72959$ $\quad \mu_5 = -1.76522$

$\mu_6 = 1.00000$

We can differentiate Equation (5.18) six times and form the derivative sequences for $u = 0$ and $v = 1$. The sign changes are summarized in Table 5.1.

Table 5.1 - Derivative Series of Fourier-Budan Theorem:
7-Component Hydrocarbon System (Standing, 1977)

	$u = 0$	$v = 1$
$f(\alpha)$	−	+
$f'(\alpha)$	+	−
$f''(\alpha)$	−	+
$f'''(\alpha)$	+	+
$f^{(4)}(\alpha)$	−	−
$f^{(5)}(\alpha)$	−	−
$f^{(6)}(\alpha)$	+	+
	$A = 5$	$B = 4$

Since $A - B = 1$, there exists only a single root of the polynomial on the interval [0,1]; an examination of the graph of the function (Figure 5.3) confirms this fact. Therefore, we can use the interval-halving and Newton-Raphson solution procedures outlined at the beginning of this chapter with confidence that they will obtain a unique, realistic value of the vapor fraction.

5.2.5 Decenary System

Gregory et al. (1971) performed equilibrium measurements on a lean natural gas at cryogenic conditions. It is reported as a ten-component system with the feed composition shown in Table 5.2. This is a very "sparse" ten-component gas, with six components present in extremely dilute concentrations. The K-values for the last four constituents were zero for eleven of the sixteen operating conditions tested in this work, denoted in Table 5.3 by the run number assigned by the investigators. The remaining twelve sets of published data duplicate conditions in one of the tested runs or are incomplete due to apparatus failure. The use of the Fourier-Budan theorem provides warning that perhaps this gas would be better represented by an equivalent "lumped" system.

Recall that the number of roots predicted by the Fourier-Budan theorem is the maximum possible and may differ from the true value by only an even integer. This is demonstrated in Table 5.4, where both the predicted and actual number of roots for each run are tabulated. The Newton-Raphson technique converges to the experimental value for ten of the sixteen runs. Three other data points follow the proper trend, while no root is found on the interval [0,1] for three other conditions (Figure 5.4).

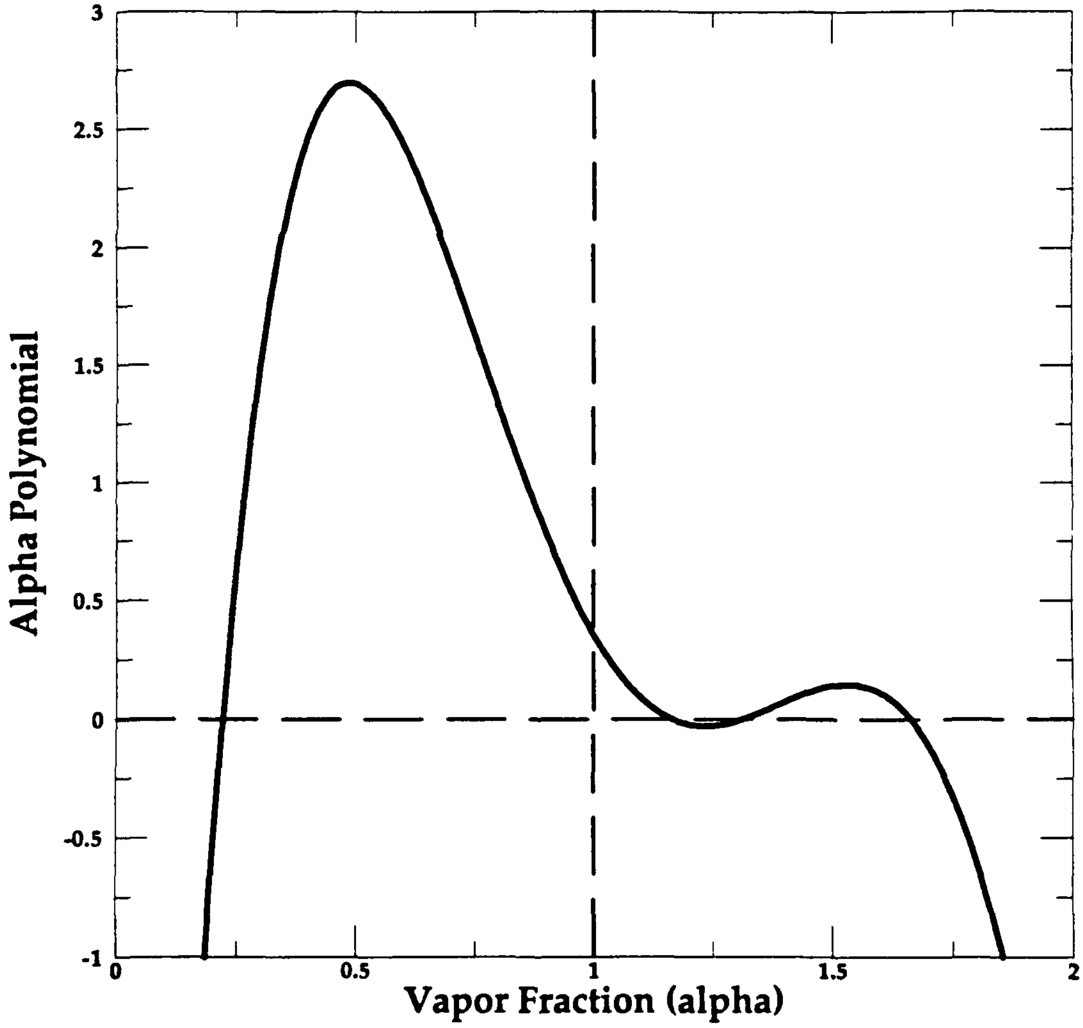


Figure 5.3: The Polynomial for a 7-Component Hydrocarbon System

Table 5.2 - Feed Composition: 10-Component Natural Gas (Gregory et al., 1971)

Component	z_i	Component	z_i
Nitrogen	0.00600	*n*-Butane	0.00070
Methane	0.95970	*i*-Pentane	0.00030
Ethane	0.03000	*n*-Pentane	0.00010
Propane	0.00390	3-Methylpentane	0.00025
i-Butane	0.00070	2-Methylhexane	0.00015

Table 5.3 - Experimental Flash Conditions: 10-Component Natural Gas (Gregory et al., 1971)

Run	Pressure (psia)	Temperature (°F)	Run	Pressure (psia)	Temperature (°F)
1	300.0	-156.3	14	100.0	-200.0
3	100.0	-206.0	15	500.6	-127.0
4	700.0	-103.0	18	23.0	-252.0
7	500.0	-125.0	20	497.0	-129.0
8	498.5	-120.0	21	23.5	-251.5
9	695.0	-105.0	25	700.0	-107.0
10	100.0	-203.3	26	298.0	-157.5
12	100.0	-195.0	28	500.0	-130.0

Table 5.4 - Results of the α-Polynomial and Fourier-Budan Theorem: 10-Component Natural Gas (Gregory et al., 1971)

Run	Root Limit (Actual)	Newton-Raphson Iterations	Roots on [0,1]		
			Initial Guess	*Calculated*	Experimental
1	2	3 3	0.605 0.825	0.603 0.822	0.603 0.603
3	2 (0)	*	*****	*****	0.155
4	2 (0)	*	*****	*****	0.911
7	3 (1)	4	0.775	0.772	0.761
8	1	3	0.905	0.904	0.908
9	2 (0)	*	*****	*****	0.795
10	4 (2)	4 4	0.695 0.915	0.692 0.912	0.687 0.687
12	2	4 5	0.895 0.965	0.891 0.966	0.890 0.890
14	3 (1)	4	0.835	0.830	0.830
15	2	3 3	0.735 0.845	0.737 0.843	0.747 0.747
18	4 (2)	3 3	0.045 0.415	0.044 0.415	0.109 0.109
20	4 (2)	3 3	0.585 0.775	0.587 0.773	0.591 0.591
21	2	4 3	0.015 0.645	0.011 0.642	0.078 0.078
25	1	3	0.385	0.380	0.548
26	5 (1)	3	0.435	0.434	0.430
28	3 (1)	4	0.485	0.480	0.486

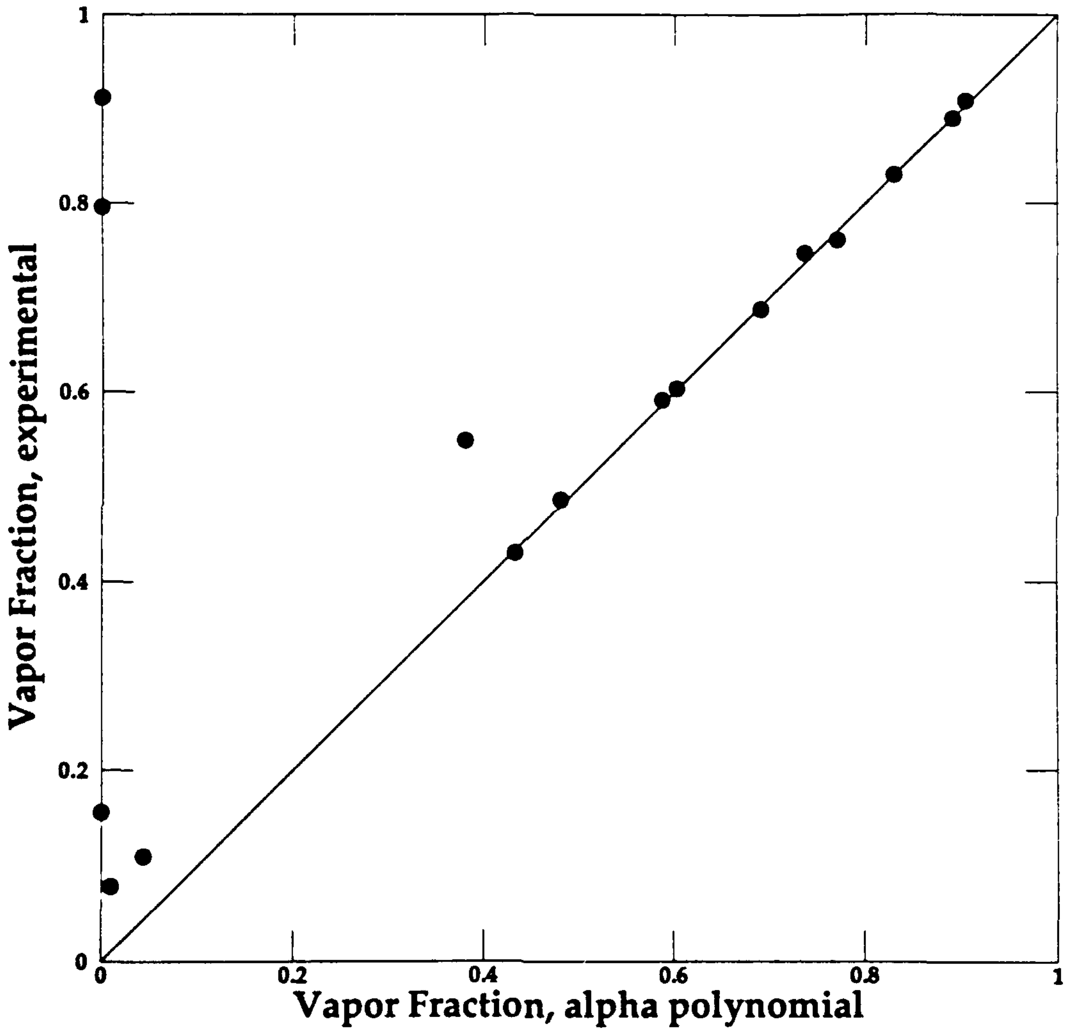


Figure 5.4: A Comparison of Predicted and Experimental Vapor Fractions for a 10-Component Natural Gas

An examination of graphs of the polynomial's behavior over a range of α for Runs 3 and 4 (Figures 5.5 and 5.6) confirms the algorithm's prediction that no roots exist within the phase envelope. The case of Run 9 is not so obvious. Its graph (Figure 5.7) shows that the function exists entirely above the abscissa; hence no root is possible. However, if the resolution of the graph is increased to examine the region very near the axis, two local minima are revealed (Figure 5.8). One of these corresponds to the experimental value of α determined for this run. The polynomial is attempting to represent the system's behavior but is not completely successful because the low concentration of certain components effectively prevents the gas from acting like a decenary system.

It is instructive to compare the form of the α-polynomial with that of the Rachford-Rice objective function which is superimposed on Figure 5.7. The same high-resolution scan of the graph of the latter equation depicts no equivalent maxima which might identify the vapor fraction in the manner of the polynomial.

5.2.6 Lumping a Decenary System into a Quaternary System

The α-polynomial successfully converges to the proper answer for a majority of the runs; however, it also yields multiple roots where the physics of the problem allows only one. This suggests that the system is not being properly modeled. The categorization of the fluid as a ten-component natural gas is overly generous in light of the fact that six of its chemical constituents are present in mole fractions measured in the ten-thousandths. It was decided to represent this sparse gas as a four-component lumped system, consisting of methane, ethane, nitrogen and propane.

The mole fractions of this new fluid are normalized and the resulting cubic polynomial in α is solved. The Fourier-Budan theorem predicts a maximum of one

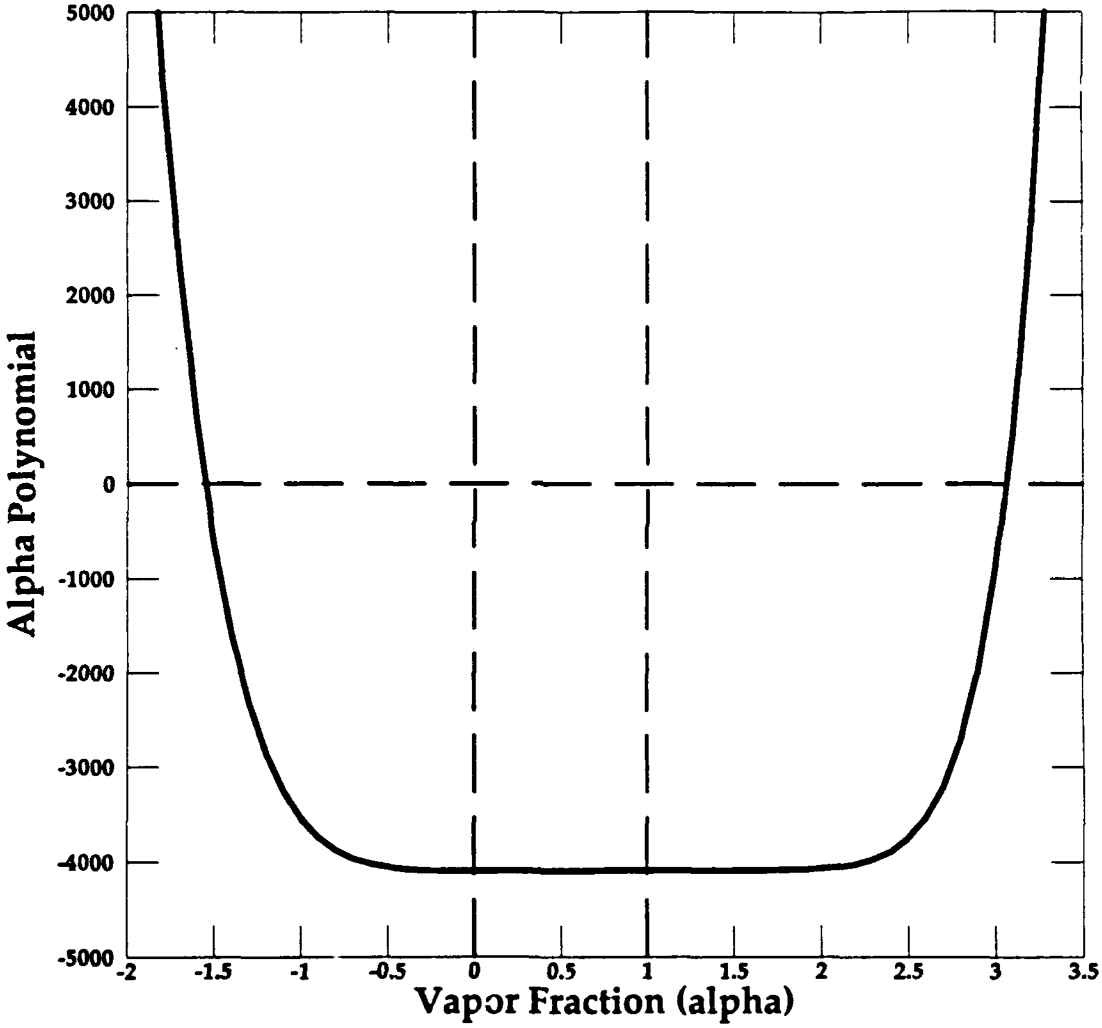


Figure 5.5: The Polynomial for a 10-Component Natural Gas [Run 3]

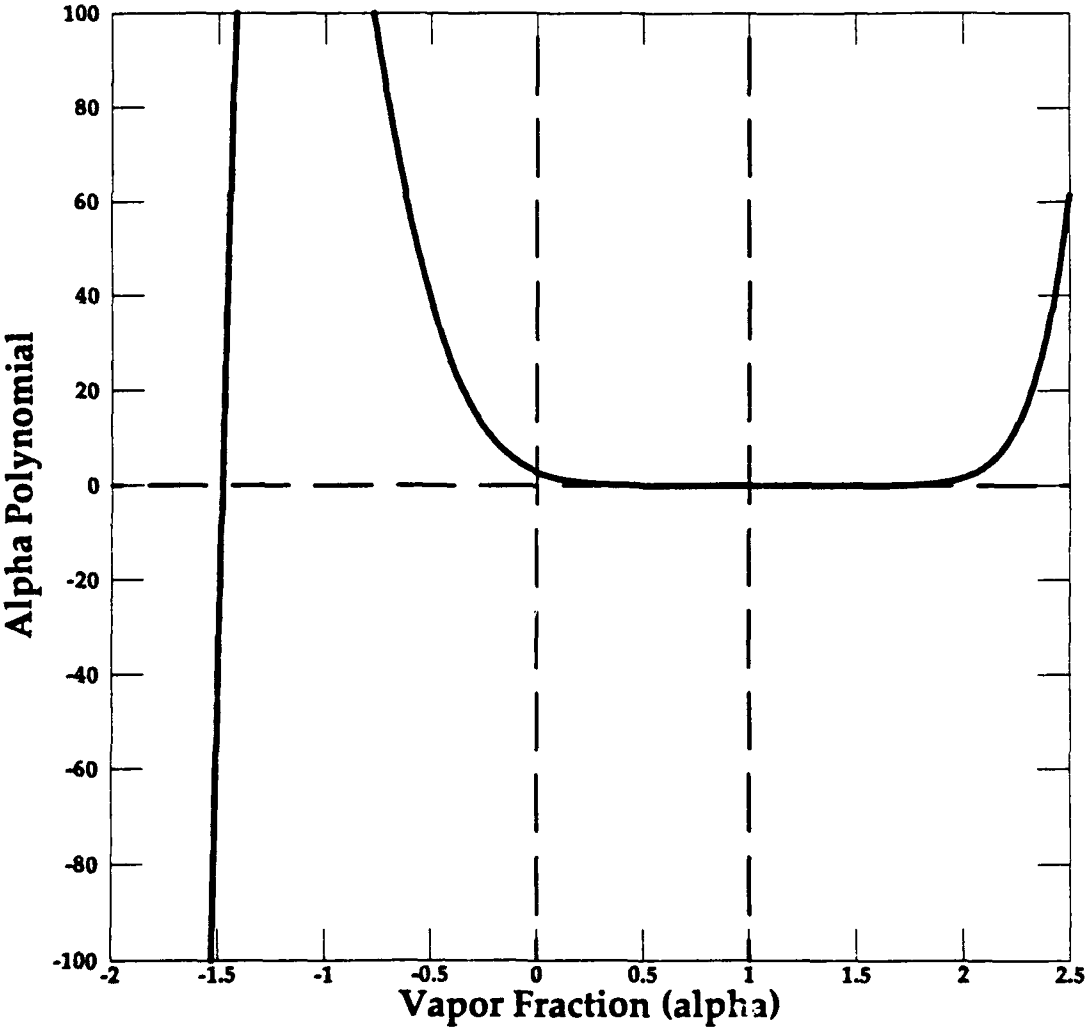


Figure 5.6: The Polynomial for a 10-Component Natural Gas [Run 4]

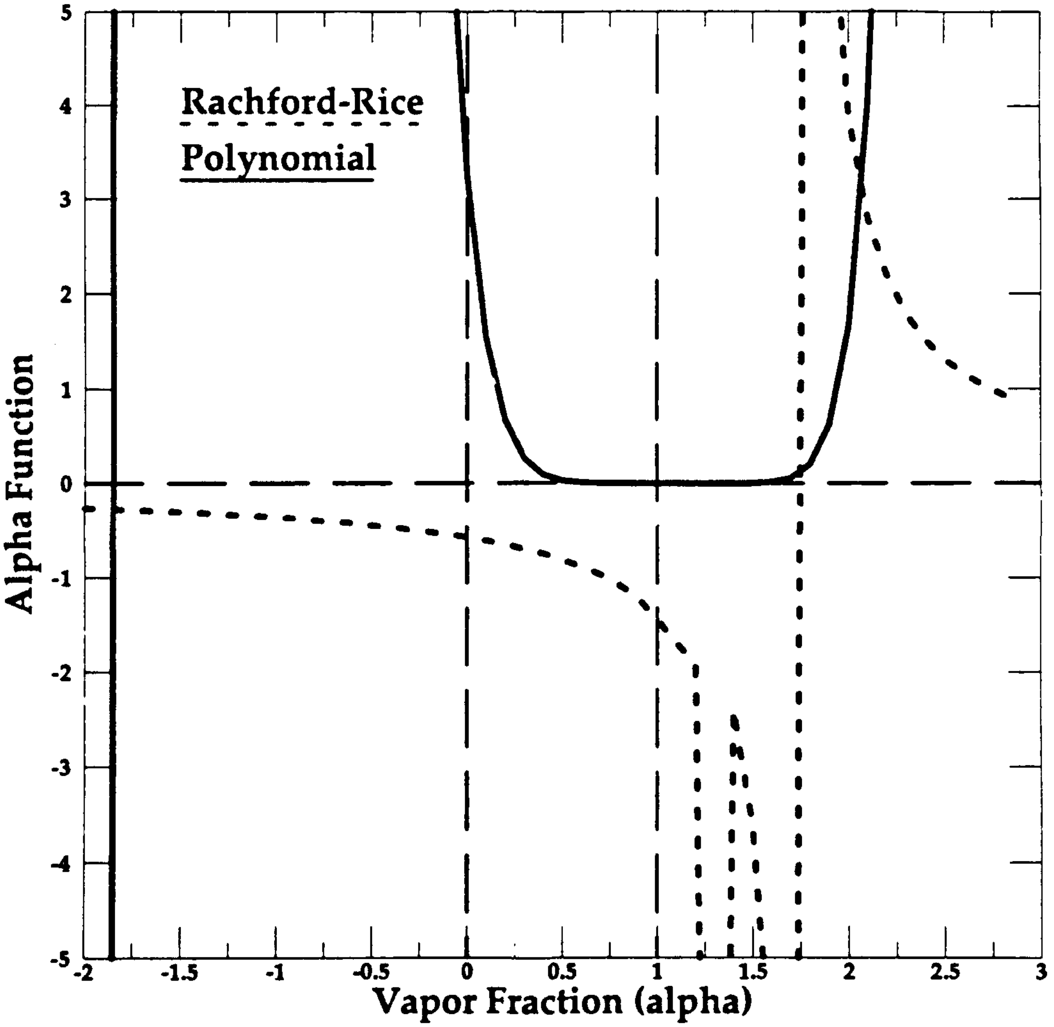


Figure 5.7: The Polynomial for a 10-Component Natural Gas [Run 9]

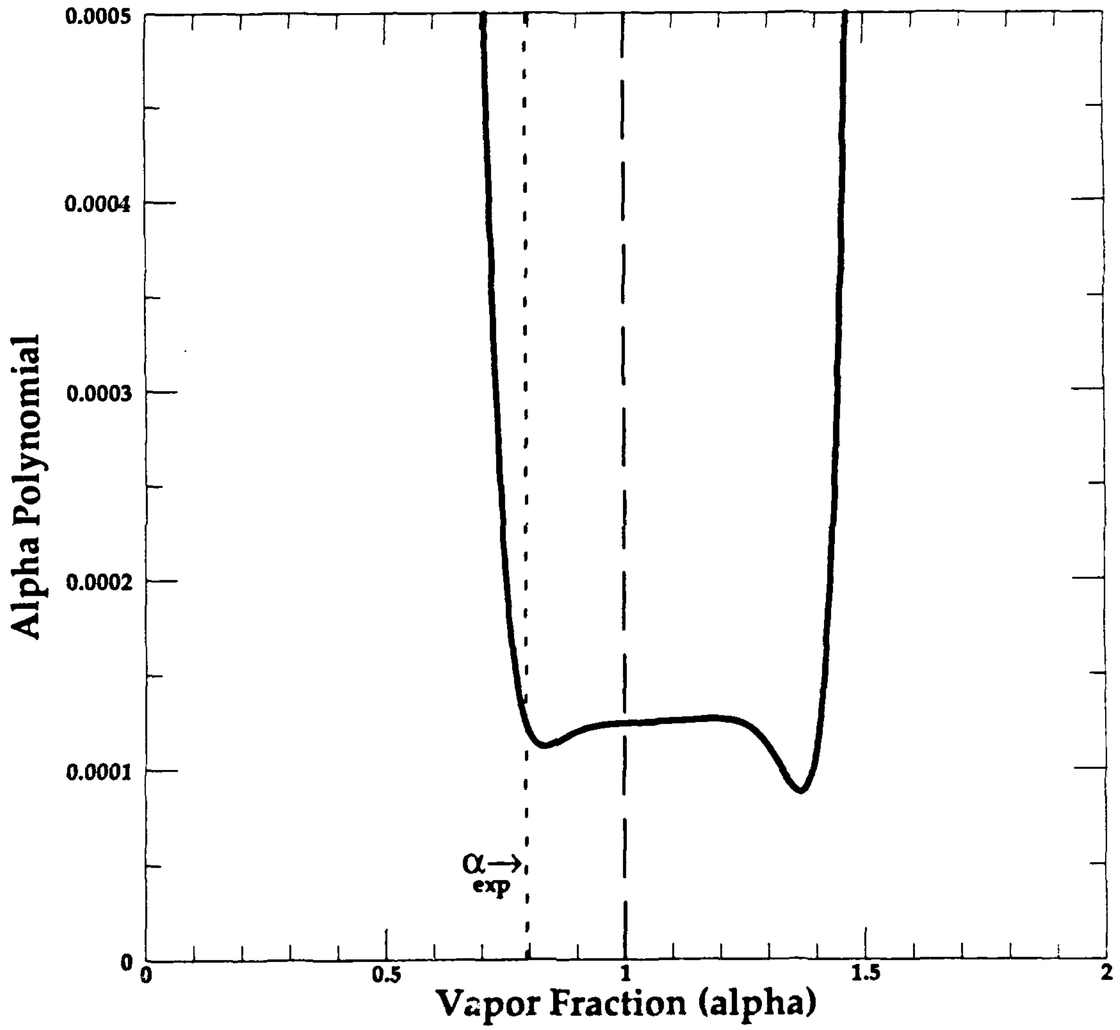


Figure 5.8: A High-Resolution View of the Polynomial for a 10-Component Natural Gas [Run 9]

root in the two-phase region, to which all sixteen runs converge. The numbers tabulated in Table 5.5 and displayed graphically in Figure 5.9 attest to the validity of this lumping scheme. An attempt was made to eliminate the next leanest component--propane--from the mixture and model the system as a ternary, but this resulted in spurious roots for all data runs and was hence rejected as unrealistic.

Table 5.5 - Results of the α-Polynomial and Fourier-Budan Theorem:
"Lumped" 4-Component Natural Gas
(Gregory et al., 1971)

Run	Root Limit (Actual)	Newton-Raphson Iterations	Roots on [0,1]		
			Initial Guess	Calculated	Experimental
1	1	3	0.635	0.631	0.603
3	1	3	0.035	0.038	0.155
4	1	3	0.995	0.998	0.911
7	1	3	0.815	0.819	0.761
8	1	3	0.935	0.932	0.908
9	1	3	0.925	0.920	0.795
10	1	3	0.715	0.711	0.687
12	1	3	0.895	0.899	0.890
14	1	3	0.845	0.842	0.830
15	1	3	0.765	0.770	0.747
18	1	3	0.035	0.037	0.109
20	1	3	0.635	0.631	0.591
21	1	3	0.015	0.012	0.078
25	1	3	0.775	0.779	0.548
26	1	3	0.475	0.473	0.430
28	1	3	0.535	0.536	0.486

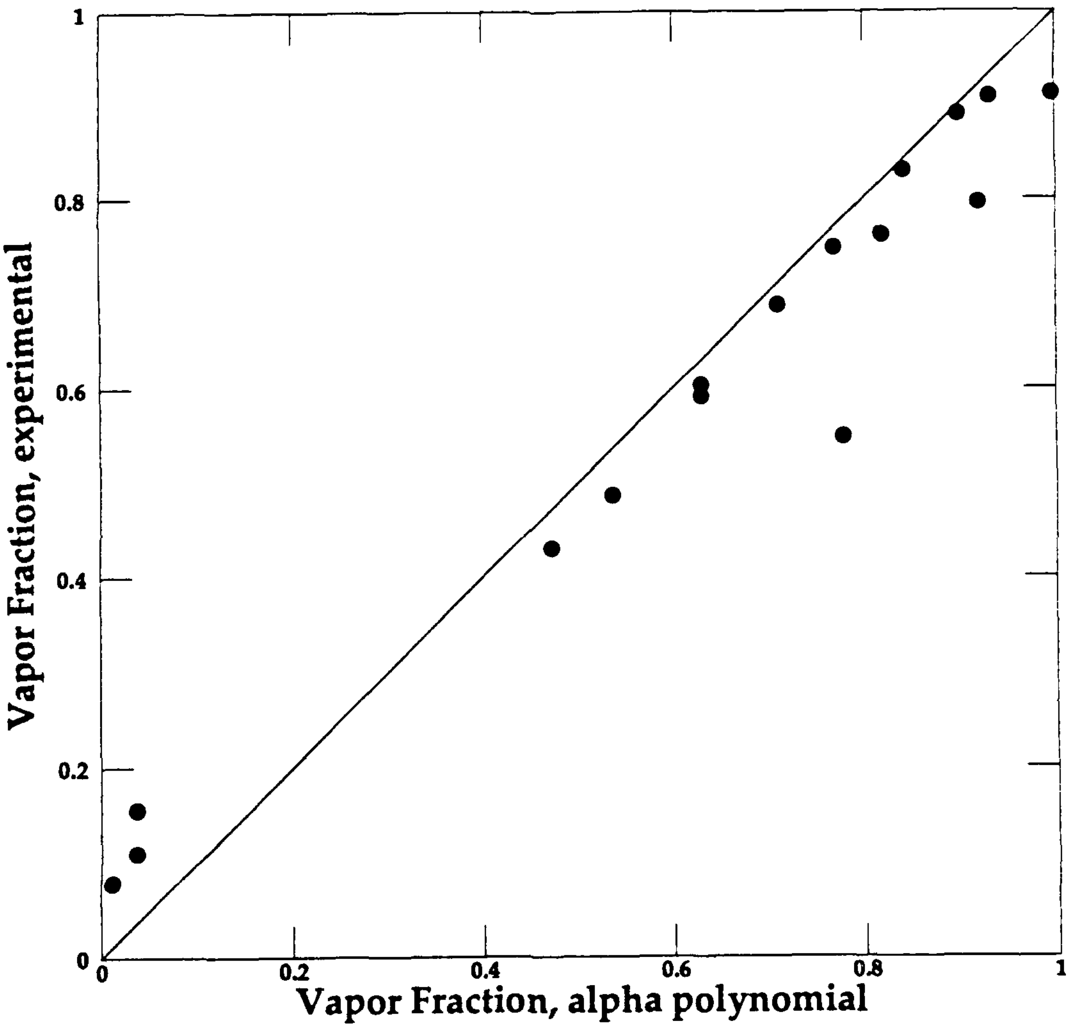


Figure 5.9: A Comparison of Predicted and Experimental Vapor Fractions for a Lumped 4-Component Natural Gas

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

1. The Rachford-Rice objective function can be represented as a polynomial in α, the system vapor fraction. Its coefficients involve elementary symmetric functions, which can be expressed in terms of a determinant whose elements are functions of equilibrium ratios and feed composition.

2. The polynomial has been shown to be well-behaved within the two-phase vapor-liquid region if the system is properly defined in terms of the number of its components. The vapor fraction root on the interval [0,1] can be quickly determined using an ordinary interval-halving technique to provide an initial estimate to the Newton-Raphson iterative method.

3. The regular behavior of the polynomial lends itself to use as a descriptive tool for the conditions of the system within the phase envelope. The Rachford-Rice objective function is not capable of this task as evidenced by Figure 5.7; its unpredictable, singular nature offers no clue to the reason a root was not found on the interval [0,1] for this case. As discussed earlier, a close examination of the curve of the polynomial revealed a local minimum at the experimental value of α. This became a realistic root ($\alpha < 1$) once the system was lumped into four components.

4. The theory of polynomials is well-developed and their behavior and zeros can be predicted with good confidence. By the use of the Fourier-Budan theorem, it can be shown mathematically that only one real root for the α-polynomial can exist on the interval [0,1] for a *well-defined* system. This eliminates the need to solve for all the roots of an N-th order polynomial.

5. The Fourier-Budan theorem can be used as a tool for investigating various lumping schemes whereby multicomponent fluids are condensed to equivalent systems composed of fewer components. The phase behavior of sparse fluids having dilute concentrations of several constituents does not seem to be well-described by the polynomial of degree appropriate to the number of components. In this case, the polynomial yields no roots or at least two roots inside the phase envelope for certain temperature and pressure conditions. It appears that a lumping scheme can be tuned by generating pseudocomponents to give successive polynomials of lower degree until only one root is determined on the interval $0 \le \alpha \le 1$.

6.2 Recommendations

1. Further study should focus on coupling the polynomial algorithm to an equation of state and extending this work to flash calculations involving more than two phases.

2. Timing studies could be conducted to determine the exact savings in CPU time realized by the use of the polynomial instead of the Rachford-Rice objective function.

3. Peng et al. (1975) estimate that 75% of the total computing time in compositional reservoir simulation may be related to the phase-behavior part of the program. The savings in computational workload realized by the use of the generalized equation developed in this work might be applied to the employment of an EOS better able to predict fluid thermodynamic properties. The highly nonlinear nature of the equations proposed by Benedict, Webb and Rubin (1940, 1942, 1951) or Lee and Kessler (1975) require iterative solutions but they yield much more accurate representations of fluid behavior, especially of nonhydrocarbon systems.

4. Since the coefficients of the generalized polynomial depend only on the feed composition and equilibrium ratios, research should continue to develop highly accurate K-value prediction methods (e.g., on the basis of convergence pressure). If this can be done with sufficient accuracy, the fugacity-convergence approach and its inherent dependence on an equation of state can be supplanted for flash calculations where nothing more than the phase split and compositions are required. The polynomial algorithm can be solved on a programmable scientific calculator and would provide the engineer with a valuable predictive tool in situations where he or she has no access to a computer capable of running an EOS-based flash routine.

BIBLIOGRAPHY

Abbott, M.M., "Cubic Equations of State: An Interpretative Review," *ACS Advances in Chemistry Series* (1979), **182**, 47-70.

Abhvani, A.S., Beaumont, D.N., "Development of an Efficient Algorithm for the Calculation of Two-Phase Flash Equilibria," *SPE Reservoir Engineering* (Nov 1987), **2**, 695-701.

Asselineau, L., Bogdanic, G., Vidal, J., "A Versatile Algorithm for Calculating Vapour-Liquid Equilibria," *Fluid Phase Equilibria* (Nov 1979), **3**, 273-290.

Baker, L.E., Luks, K.D., "Critical Point and Saturation Pressure Calculations for Multipoint Systems," *SPE Journal* (Feb 1980), **20**, 15-24.

Barbeau, E.J., Polynomials, Springer-Verlag, New York, 1989.

Benedict, M., Webb, G.B., Rubin, L.C., "An Empirical Equation for Thermodynamic Properties of Light Hydrocarbons and Their Mixtures," *Journal of Chemical Physics* (Apr 1940), **8**, 334-345; (Dec 1942), **9**, 747-758.

Benedict, M., Webb, G.B., Rubin, L.C., "An Empirical Equation for Thermodynamic Properties of Light Hydrocarbons and Their Mixtures: Fugacities and Liquid-Vapor Equilibria," *Chemical Engineering Progress* (Aug 1951), **47**, 419; (Sep 1951), **47**, 449-454.

Benmekki, E.-H., "Prediction of the Entire Range of Vapor-Liquid Equilibrium States," M.S. thesis, Illinois Institute of Technology, Chicago, IL, August 1984.

Bloomer, O.T., Gami, D.C., Parent, J.D., "Physical-Chemical Properties of Methane-Ethane Mixtures," Research Bulletin 22, Institute of Gas Technology, Illinois Institute of Technology, Chicago, IL, July 1953.

Boston, J.F., Britt, H.I., "A Radically Different Formulation and Solution of the Single-Stage Flash Problem," *Computers and Chem. Eng.* (1978), **2**, 109-122.

Brinkman, F.H., Sicking, J.N., "Equilibrium Ratios for Reservoir Studies," *Petroleum Transactions, AIME* (1960), **219**, 313-319.

Cox, E.R., "Pressure-Temperature Chart for Hydrocarbon Vapors," *Ind. Eng. Chem.* (Jun 1923), **15**, 592-593.

David, F.N., Kendall, M.G., Barton, D.E., Symmetric Function and Allied Tables, Cambridge University Press, Cambridge, England, 1966.

Dodson, C.R., Standing, M.B.,"Prediction of Volumetric and Phase Behavior of Naturally Occurring Hydrocarbon Systems," *API Drilling and Production Practice* (1941), 326.

Edmister, W.C., "Applications of Thermodynamics to Hydrocarbon Processing: Part XXII--Convergence Correction to Vapor-Liquid Equilibrium Ratios," *Petroleum Refiner* (Sep 1949), **28**, 95-102.

Enick, R.M., Holder, G.D., Mohamed, R., "Four-Phase Flash Equilibrium Calculations Using the Peng-Robinson Equation of State and a Mixing Rule for Asymmetric Systems," *SPE Reservoir Engineering* (Nov 1987), **2**, 687-694.

Fussell, D.D., Yanosik, J.L., "An Iterative Sequence for Phase-Equilibria Calculation Incorporating the Soave-Redlich-Kwong Equation of State," *SPE Journal* (Jun 1978), **18**, 173-182.

Fussell, L.T., "A Technique for Calculating Multiphase Equilibria," *SPE Journal* (Aug 1979), **19**, 203-210.

Gosset, R., Heyen, G., Kalitventzeff, B., "An Efficient Algorithm to Solve Cubic Equations of State," *Fluid Phase Equilibria* (Jan 1986), **25**, 51-64.

Gregory, D.P., Djordjevich, L., Kao, R., Anderson, G.L., "Thermodynamic Properties of a Lean Natural Gas at Cryogenic Conditions," Technical Report 11, Institute of Gas Technology, Illinois Institute of Technology, Chicago, IL, June 1971.

Hadden, S.T., "Vapor-Liquid Equilibria in Hydrocarbon Systems," *Chemical Engineering Progress* (Jan 1948), **44**, 37-54, (Feb 1948), 135-156.

Hadden, S.T., "Convergence Pressure in Hydrocarbon Vapor-Liquid Equilibria," *Chemical Engineering Progress Symposium Series* (1953), **49**, 53-66.

Hanson, G.H., Brown, G.G., "Vapor-Liquid Equilibria in Mixtures of Volatile Paraffins," *Ind. Eng. Chem.* (Sep 1945), **37**, 821-825.

Hoffmann, A.E., Crump, J.S., Hocott, C.R., "Equilibrium Constants for a Gas-Condensate System," *Petroleum Transactions, AIME* (1953), **198**, 1-10.

Huron, M.-J., Vidal, J., "New Mixing Rules in Simple Equations of State for Representing Vapor-Liquid Equilibria of Strongly Non-Ideal Mixtures," *Fluid Phase Equilibria* (1979), **3**, 255-271.

Jhaveri, B.S., Youngren, G.K., "Three-Parameter Modification of the Peng-Robinson Equation of State to Improve Volumetric Predictions," *SPE Reservoir Engineering* (Aug 1988), **3**, 1033-1040.

Katz, D.L., Brown, G.G., "Vapor Pressure and Vaporization of Petroleum Fractions," *Ind. Eng. Chem. Res.* (Dec 1933), **25**, 1373-1384.

Katz, D.L., Hachmuth, K.H., "Vaporization Equilibrium Constants in a Crude Oil-Natural Gas System," *Ind. Eng. Chem. Res.* (Sep 1937), **29**, 1072-1077.

Katz, D.L., Kurata, F., "Retrograde Condensation," *Ind. Eng. Chem. Res.* (Jun 1940), **32**, 817-827.

Kay, W.B., "Density of Hydrocarbon Gases and Vapors," *Ind. Eng. Chem. Res.* (Sep 1936), **28**, 1014-1019.

Knapp, H., Döring, R., Oellrich, L., Plöcker, U., Prausnitz, J.M., Vapor-Liquid Equilibria for Mixtures of Low Boiling Substances, DECHEMA Chemistry Data Series, vol. VI, DECHEMA Deutsche Gesellschaft für Chemisches Apparatewesen, Frankfurt/Main, Federal Republic of Germany, 1982.

Kumar, S., Gas Production Engineering, Gulf Publishing Co., Houston, 1987.

Lawal, A.S., "Consistent Rule for Selecting Roots in Cubic Equations of State," *Ind. Eng. Chem. Res.* (Apr 1987), **26**, 857-859.

Lee, B.I., Kesler, M.G., "A Generalized Thermodynamic Correlation Based on Three-Parameter Corresponding States," *AIChE Journal* (May 1975), **21**, 510-527.

Macdonald, I.G., Symmetric Functions and Hall Polynomials, Oxford University Press, Oxford, England, 1979.

MacMahon, P.A., An Introduction to Combinatory Analysis, Cambridge University Press, London, England, 1920.

Martin, J.J., "Cubic Equations of State--Which?" *Ind. Eng. Chem. Fundam.* (May 1979), **18**, 81-97.

Mathias, P.M., Benson, M.S., "Computational Aspects of Equations of State: Fact and Fiction," *AIChE Journal* (Dec 1986), **32**, 2087-2090.

Mathias, P.M., Copeman, T. W., "Extension of the Peng-Robinson Equation of State to Complex Mixtures: Evaluation of the Various Forms of the Local Composition Concept," *Fluid Phase Equilibria* (Oct 1983), **13**, 91-108.

Mathias, P.M., Naheiri, T., Oh, E.M., "A Density Correction for the Peng-Robinson Equation of State," *Fluid Phase Equilibria* (Jul 1989), **47**, 77-87.

Mehra, R.K., Heidemann, R.A., Aziz, K., "Computation of Multiphase Equilibrium for Compositional Simulation," *SPE Journal* (Feb 1982), **22**, 61-68.

Mehra, R.K., Heidemann, R.A., Aziz, K., "An Accelerated Successive Substitution Algorithm," *Canadian Journal of Chemical Engineering* (Aug 1983), **61**, 590-596.

Michelsen, M.L., "The Isothermal Flash Problem. Part II: Phase-Split Calculation," *Fluid Phase Equilibria* (Sep 1982), **9**, 21-40.

Mott, R.E., "Development and Evaluation of a Method for Calculating Phase Behaviour of Multi-Component Hydrocarbon Mixtures Using an Equation of State," U.K. Atomic Energy Authority, Dorchester (Feb 1980), AEEW-R 1331.

Mott, R.E., "The Winfrith PVT Package," U.K. Atomic Energy Authority, Dorchester (Apr 1983), AEEW-R 1636.

Muskat, M., Physical Principles of Oil Production, McGraw-Hill, New York, 1949.

Nghiem, L.X., Aziz, K., Li, Y.K., "A Robust Iterative Method for Flash Calculations Using the Soave-Redlich-Kwong or the Peng-Robinson Equation of State," *SPE Journal* (Jun 1983), **23**, 521-530.

Parikh, J.S., Bukacek, R.F., Graham, L., Leipziger, S., "Dew and Bubble Point Measurements for a Methane-Ethane-Propane Mixture," *Journal of Chemical and Engineering Data* (1984), **29**, 301-303.

Péneloux, A., Rauzy, E., Fréze, R., "A Consistent Correction for Redlich-Kwong-Soave Volumes," *Fluid Phase Equilibria* (Mar 1982), **8**, 7-23.

Peng, D.-Y., "An Empirical Method for Calculating Vapor-Liquid Critical Points of Multicomponent Mixtures," *Canadian Journal of Chemical Engineering* (Oct 1986), **64**, 827-830.

Peng, D.-Y., Robinson, D.B., "A New Two-Constant Equation of State," *Ind. Eng. Chem. Fundam.* (Feb 1976a), **15**, 59-64.

Peng, D.-Y., Robinson, D.B., "Two and Three Phase Equilibrium Calculations for Systems Containing Water," *Canadian Journal of Chemical Engineering* (Dec 1976b), **54**, 595-599.

Peng, D.-Y., Robinson, D.B., "A Rigorous Method for Predicting the Critical Properties of Multicomponent Systems from an Equation of State," *AIChE Journal* (Mar 1977), **23**, 137-144.

Peng, D.-Y., Robinson, D.B., Bishnoi, P.R., "The Use of the Soave-Redlich-Kwong Equation of State for Predicting Condensate Fluid Behavior," *Proceedings*, Ninth World Petroleum Congress, Tokyo (1975), 377-388.

Pitzer, K.S., Lippmann, D.Z., Curl, R.F. Jr., Huggins, C.M., Petersen, D.E., "The Volumetric and Thermodynamic Properties of Fluids. II. Compressibility Factor, Vapor Pressure and Entropy of Vaporization," *Journal of Amer. Chem. Soc.* (5 Jul 1955), **77**, 3433-3440.

Poling, B., Grens, E.A. II, Prausnitz, J.M., "Thermodynamic Properties from a Cubic Equation of State: Avoiding Trivial Roots and Spurious Derivatives," *Ind. Eng. Chem. Proc. Des. Dev.* (Jan 1981), **20**, 127-130.

Proust, P., Vera, J.H., "PRSV: The Stryjek-Vera Modification of the Peng-Robinson Equation of State. Parameters for Other Pure Compounds of Industrial Interest," *Canadian Journal of Chemical Engineering* (Feb 1989), **67**, 170-173.

Rachford, H.H. Jr., Rice, J.D., "Procedure for Use of Electronic Digital Computers in Calculating Flash Vaporization Hydrocarbon Equilibrium," *Petroleum Transactions, AIME* (1952), **195**, 327-328.

Redlich, O., Kwong, J.N.S., "On the Thermodynamics of Solutions V. An Equation of State. Fugacities of Gaseous Solutions," *Chemical Reviews* (1949), **44**, 233-244.

Reid, R.C., Prausnitz, J.M., Poling, B.E., The Properties of Gases and Liquids, 4th ed., McGraw-Hill, New York, 1987.

Risnes, R., Dalen, V., "Equilibrium Calculations for Coexisting Liquid Phases," *SPE Journal* (Feb 1984), **24**, 87-96.

Risnes, R., Dalen, V., Jensen, J.I., "Phase Equilibrium Calculations in the Near-Critical Region," *Proceedings*, European Symposium on EOR, Bournemouth (1981), 329-350.

Robinson, D.B., "Author's Reply to Comments on 'A New Two-Constant Equation of State'," *Ind. Eng. Chem. Fundam.* (May 1979), **18**, 197.

Robinson, D.B., Peng, D.-Y., "The Characterization of the Heptanes and Heavier Fractions for the GPA Peng-Robinson Programs," Research Report RR-28, Gas Processors Association, Tulsa, OK, March 1978.

Robinson, D.B., Peng, D.-Y., Chung, S.Y.-K., "The Development of the Peng-Robinson Equation and its Application to Phase Equilibrium in a System Containing Methanol," *Fluid Phase Equilibria* (Nov 1985), **24**, 25-41.

Roe, J.R., "Interfunctional Expressibility Problems of Symmetric Functions," Ph.D. dissertation, Syracuse University, Syracuse, NY, June 1918.

Schick, L.M., Prausnitz, J.M., "On the Correlation of Critical Properties of Mixtures," *AIChE Journal* (Jul 1968), **14**, 673.

Soave, G., "Equilibrium Constants from a Modified Redlich-Kwong Equation of State," *Chemical Engineering Science* (Jun 1972), **27**, 1197-1203.

Souders, M. Jr., Selheimer, C.W., Brown, G.G., "Equilibria between Liquid and Vapor Solutions of Paraffin Hydrocarbons," *Ind. Eng. Chem. Res.* (May 1932), **24**, 517-519.

Spencer, C.F., Danner, R.P., "Improved Equation for Prediction of Saturated Liquid Density," *Journal of Chemical and Engineering Data* (1972), **17**, 236-241.

Standing, M.B., Volumetric and Phase Behavior of Oil Field Hydrocarbon Systems, SPE-AIME, Dallas, 1977.

Standing, M.B., "A Set of Equations for Computing Equilibrium Ratios of a Crude Oil/Natural Gas System at Pressures Below 1000 psia," *Journal of Petroleum Technology* (Sep 1979), 1193-1195.

Stryjek, R., Vera, J.H., "An Improved Cubic Equation of State," *American Chemical Society Symposium Series* (1986a), **300**, 561-570.

Stryjek, R., Vera, J.H., "PRSV: An Improved Peng-Robinson Equation of State for Pure Compounds and Mixtures," *Canadian Journal of Chemical Engineering* (Apr 1986b), **64**, 323-333.

Stryjek, R., Vera, J.H., "PRSV--An Improved Peng-Robinson Equation of State with New Mixing Rules for Strongly Nonideal Mixtures," *Canadian Journal of Chemical Engineering* (Apr 1986c), **64**, 334-340.

Stryjek, R., Vera, J.H., "PRSV2: A Cubic Equation of State for Accurate Vapor-Liquid Equilibria Calculations," *Canadian Journal of Chemical Engineering* (Oct 1986d), **64**, 820-826.

van der Waals, J.D., "On the Continuity of the Gaseous and Liquid States," Ph.D. dissertation, Sigthoff University, Leiden, The Netherlands, 1873.

Varotsis, N., "A Robust Prediction Method for Rapid Phase-Behavior Calculations," *SPE Reservoir Engineering* (May 1989), **4**, 237-243.

Varotsis, N., Todd, A.C., Stewart, G., "An Iterative Method for Phase Equilibria Calculations with Particular Application to Multicomponent Miscible Systems," *Proceedings*, European Symposium on EOR, Bournemouth (1981), 313-328.

Vera, J.H., Huron, M.-J., Vidal, J., "On the Flexibility and Limitations of Cubic Equations of State," *Chem. Eng. Comm.* (1984), **26**, 311-318.

Vidal, J., "Equations of State--Reworking the Old Forms," *Fluid Phase Equilibria* (Oct 1983), **13**, 15-33.

Walas, S.M., Phase Equilibria in Chemical Engineering, Butterworth Publishers, Boston, 1985.

Warren, J.H., "Explicit Determination of the Vapor Fraction in Flash Calculations," M.S. thesis, Pennsylvania State University, University Park, PA, December 1991.

White, R.R., Brown, G.G., "Phase Equilibria at High Temperatures," *Ind. Eng. Chem.* (Oct 1942), **34**, 1162-1174.

Whitson, C.H., Torp, S.B., "Evaluating Constant-Volume Depletion Data," paper SPE 10067, *Proceedings*, SPE 56th Annual Fall Technical Conference, San Antonio, TX (5-7 October 1981).

Wilczek-Vera, G., Vera, J.H., "A Comparative Study of Mixing Rules for Cubic Equations of State," *Fluid Phase Equilibria* (1987), **37**, 241-253.

Wilson, G.M., "A Modified Redlich-Kwong Equation of State, Applications to General Physical Data Calculations," paper 15C, *Proceedings*, AIChE National Meeting, Cleveland, OH (4-7 May 1969).

Winn, F.W., "Simplified Nomographic Presentation--Hydrocarbon Vapor-Liquid Equilibria," *Chemical Engineering Progress Symposium Series* (1952), **48**, 121-134.

Wu, D., Sandler, S.I., "Generalized Temperature-Dependent Parameters for the Peng-Robinson Equation of State for *n*-Alkanes," *Ind. Eng. Chem. Res.* (Jul 1989), **28**, 1103-1106.

Xu, Z., Sandler, S.I., "Temperature-Dependent Parameters and the Peng-Robinson Equation of State," *Ind. Eng. Chem. Res.* (Mar 1987a), **26**, 601-606.

Xu, Z., Sandler, S.I., "Application to Mixtures of the Peng-Robinson Equation of State with Fluid Specific Parameters," *Ind. Eng. Chem. Res.* (Jul 1987b), **26**, 1234-1238.

Zaguskin, V.L., Handbook of Numerical Methods for the Solution of Algebraic and Transcendental Equations, Pergamon Press, London, England, 1961.

Zudkevitch, D., Joffe, J., "Correlation and Prediction of Vapour-Liquid Equilibria with the Redlich-Kwong Equation of State," *AIChE Journal* (1970), **16**, 112-119.

Appendix A

MATHEMATICAL PROOF OF THE GENERALIZED EQUATION

KNOWN:

$$\sum_{i=1}^{N} z_i\, C_i \prod_{j\neq i}^{N} (1 + \alpha\, C_j) = \sum_{p=1}^{N} \alpha^{(N-p)} \sum_{i=1}^{N} z_i\, C_i \left\{ a_{N-p}(C_1,\ldots, \hat{C}_i,\ldots, C_N) \right\} \qquad \text{(A-1)}$$

POSTULATE:

$$\sum_{i=1}^{N} z_i\, C_i \prod_{j\neq i}^{N} (1 + \alpha\, C_j) = \sum_{p=1}^{N} \alpha^{(N-p)} \left\{ \prod_{k=1}^{N} C_k \sum_{j=1}^{p} \left[(-1)^{j+1}\, a_{p-j} \sum_{i=1}^{N} \frac{z_i}{C_i^{\,j-1}} \right] \right\} \text{(A-2)}$$

$$= \sum_{p=1}^{N} \alpha^{(N-p)} \sum_{i=1}^{N} z_i C_i \left\{ \prod_{k=1}^{N} C_k \sum_{j=1}^{p} \left[(-1)^{j+1} \frac{a_{p-j}}{C_i^{\,j}} \right] \right\} \text{(A-3)}$$

Prove that the coefficients of α in Equation (A-1) and Equation (A-3) are equivalent:

$$\left\{ a_{N-p}(C_1,\ldots, \hat{C}_i,\ldots, C_N) \right\} = \left\{ \prod_{k=1}^{N} C_k \left[\sum_{j=1}^{p} (-1)^{j+1} \frac{a_{p-j}}{C_i^{\,j}} \right] \right\} \qquad \text{(A-4)}$$

PROOF:

We can express the a-function as:

$$a_{N-p}(C_1,\ldots, \hat{C}_i,\ldots, C_N) = \frac{\prod_{k=1}^{N} C_k}{C_i}\, a_{p-1}(C_1^{-1},\ldots, \hat{C}_i^{-1},\ldots, C_N^{-1}) \qquad \text{(A-5)}$$

where $\dfrac{\prod_{k=1}^{N} C_k}{C_i}$ represents $(N-1)$ terms: $N-p = (N-1) - (p-1)$

Eliminate the $\hat{C}_i^{-1}$ term in the RHS of Equation (A-5) by rewriting the a-function as:

$$a_{p-1}(C_1^{-1},\ldots, \hat{C}_i^{-1},\ldots, C_N^{-1}) = a_{p-1}(C_1^{-1},\ldots, C_N^{-1}) - C_i^{-1}\, a_{p-2}(C_1^{-1},\ldots, C_{N-1}^{-1}) \text{(A-6)}$$

$$a_{p-2}(C_1^{-1},\dots,\hat{C}_i^{-1},\dots,C_N^{-1}) = a_{p-2}(C_1^{-1},\dots,C_N^{-1}) - C_i^{-1}\,a_{p-3}(C_1^{-1},\dots,C_{N-1}^{-1}) \quad \text{(A-7)}$$

$$a_{p-3}(C_1^{-1},\dots,\hat{C}_i^{-1},\dots,C_N^{-1}) = a_{p-3}(C_1^{-1},\dots,C_N^{-1}) - C_i^{-1}\,a_{p-4}(C_1^{-1},\dots,C_{N-1}^{-1}) \quad \text{(A-8)}$$

. . . .

$$a_{p-(p-1)}(C_1^{-1},\dots,\hat{C}_i^{-1},\dots,C_N^{-1}) = a_{p-(p-1)}(C_1^{-1},\dots,C_N^{-1}) - C_i^{-(p-1)}\,a_0(C_1^{-1},\dots,C_{N-1}^{-1}) \quad \text{(A-9)}$$

Recall that $a_0 = 1$ and then substitute Equations (A-7), (A-8),..., (A-9) into Equation (A-6):

$$\begin{aligned} a_{p-1}(C_1^{-1},\dots,\hat{C}_i^{-1},\dots,C_N^{-1}) = {} & a_{p-1}(C_1^{-1},\dots,C_N^{-1}) - C_i^{-1}\,a_{p-2}(C_1^{-1},\dots,C_N^{-1}) \\ & + C_i^{-2}\,a_{p-3}(C_1^{-1},\dots,C_N^{-1}) - C_i^{-3}\,a_{p-4}(C_1^{-1},\dots,C_N^{-1}) + \\ & \cdots \pm C_i^{-(p-2)}\,a_{p-(p-1)}(C_1^{-1},\dots,C_N^{-1}) \pm C_i^{-(p-1)} \end{aligned} \quad \text{(A-10)}$$

After writing the recursive form for the RHS of Equation (A-10), the equation becomes:

$$a_{p-1}(C_1^{-1},\dots,\hat{C}_i^{-1},\dots,C_N^{-1}) = \sum_{j=1}^{p} (-1)^{j+1}\,C_i^{-(j-1)}\,a_{p-j} \quad \text{(A-11)}$$

Substitute Equation (A-11) into Equation (A-5):

$$a_{N-p}(C_1,\dots,\hat{C}_i,\dots,C_N) = \frac{\prod_{k=1}^{N} C_k}{C_i}\left[\sum_{j=1}^{p} (-1)^{j+1}\,\frac{a_{p-j}}{C_i^{\,j-1}}\right] \quad \text{(A-12)}$$

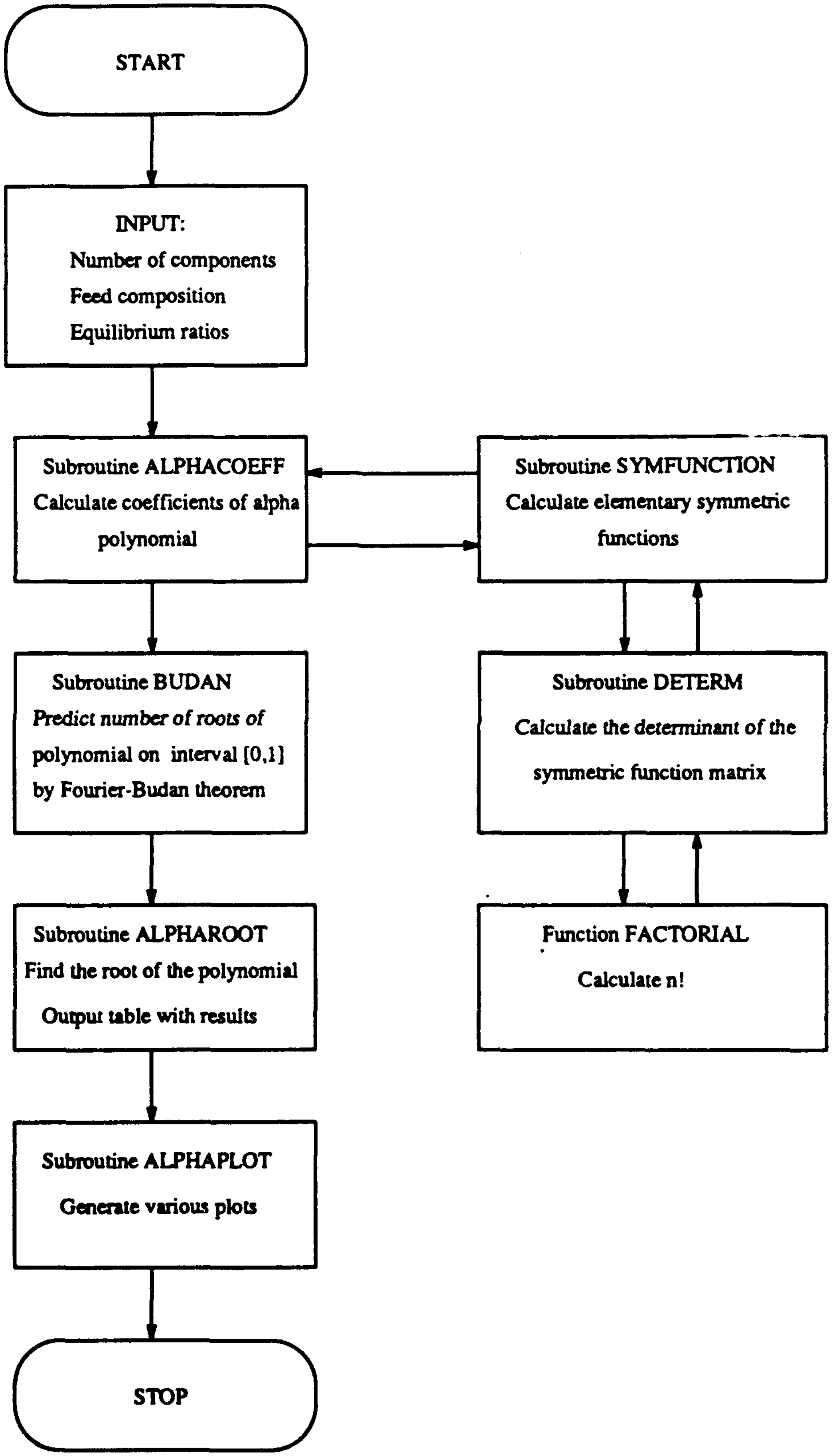
Combine C_i terms:

$$a_{N-p}(C_1,\dots,\hat{C}_i,\dots,C_N) = \prod_{k=1}^{N} C_k\left[\sum_{j=1}^{p} (-1)^{j+1}\,\frac{a_{p-j}}{C_i^{\,j}}\right] \quad \text{(A-13)}$$

Equation (A–13) = Equation (A–4) Q.E.D.

Appendix B

ALGORITHM FLOWCHART


START
INPUT:
Number of components
Feed composition
Equilibrium ratios
Subroutine ALPHACOEFF
Calculate coefficients of alpha polynomial
Subroutine SYMFUNCTION
Calculate elementary symmetric functions
Subroutine BUDAN
Predict number of roots of polynomial on interval [0,1] by Fourier-Budan theorem
Subroutine DETERM
Calculate the determinant of the symmetric function matrix
Subroutine ALPHAROOT
Find the root of the polynomial
Output table with results
Function FACTORIAL
Calculate n!
Subroutine ALPHAPLOT
Generate various plots
STOP

Appendix C
COMPUTER CODE

```
***********************************************************************
*
*                              7 Dec 91
*                          BRETT D. WEIGLE
*              Petroleum and Natural Gas Engineering Section
*                    Mineral Engineering Department
*                 College of Earth and Mineral Sciences
*                   The Pennsylvania State University
*                    University Park, Pennsylvania
*
*                             M.S. thesis
*                   Advisor:  Dr. Michael A. Adewumi
*
***********************************************************************
*                     Program ALPHATEST (FORTRAN 77)
***********************************************************************
*
*         This program calculates values of the vapor fraction,
*         given equilibrium ratios, Ki, and feed mole fractions,
*        zi.  It can be used to reproduce experimental results of
*         equilibrium flashes.
*
*         ALPHATEST calls ALPHACOEFF, BUDAN, ALPHAPLOT, and
*                             ALPHAROOT
*         ALPHACOEFF calls subroutine SYMFUNCTION
*         SYMFUNCTION calls subroutine DETERM and function FACTOR
*
*         VARIABLES:  alpha = calculated system vapor fraction
*                      beta = experimental system liquid fraction
*                      coefficient = coefficient of alpha polynomial
*                      Ki = equilibrium ratio for component i
*                      molefrac = feed mole fraction of component i
*                      Ncomp = number of components in feed
*                      Npress = number of data sets to be evaluated
*                      Pi = system pressure, psia
*                      Ti = system temperature, F
*                      xalpha = experimental system vapor fraction
*
*         It is formatted to input zi, temperature, pressure, liquid
*         mole fraction, and Ki
*
***********************************************************************

      IMPLICIT REAL*8(a-h,o-z)
      REAL*8 Ki(500,100),molefrac(0:100)
      PARAMETER(Npress=16,Ncomp=10)
      DIMENSION alpha(500), beta(500), coefficient(0:100),
     @             Pi(500), Ti(500), tarray(2), xalpha(500)
*
*        Data Input
*

*       The number of components (Ncomp) and the number of data sets
*        to be run (Npress) are specified as PARAMETERs
```

```
*
*          Open and Rewind Input and Output Files
*
      OPEN(unit=1,file='indata',status='old')
      OPEN(unit=7,file='table',status='unknown')
      OPEN(unit=8,file='plot',status='unknown')

      REWIND(unit=1)
      REWIND(unit=7)
      REWIND(unit=8)

      read(1,*) (molefrac(i), i = 1, Ncomp)
      do 1000 j = 1, Npress

         read(1,*) Pi(j), Ti(j), beta(j)
         read(1,*) (Ki(j,i), i = 1, Ncomp)
         xalpha(j) = 1.d0 - beta(j)

 1000 continue
*
*          Choose between single or multiple runs
*
      write(6,*) 'Evaluate one data set? enter 1'
      write(6,*) 'Evaluate all data sets? enter 2'
      read(5,*) numsets

      if(numsets .EQ. 1) then
         write(6,*) 'Enter number of data set for this run'
         read(5,*) j
         go to 2100
      end if

      do 2000 j = 1, Npress

 2100    write(7,*) ' '
         write(7,*) ' '
         write(7,*) '                              RUN 'j
         write(6,*) 'J = ', j
         write(7,2500) Pi(j),Ti(j),beta(j)
 2500    format('Pressure = ',f6.1,' psia   Temperature = ',f6.1,' F
     @Liquid Mole Fraction = ',f6.4)

*
*          Call subroutines
*
*          Calculate coefficients of polynomial
*
           call ALPHACOEFF(Ncomp,Npress,j,molefrac,Ki,coefficient)

*
*          Predict the number of roots on [0,1] by Fourier-Budan theorem
*
```

```
            call BUDAN(j,Ncomp,coefficient,numroot)

*
*           Solve for the roots by Newton-Raphson method
*
            call ALPHAROOT(j,Ncomp,coefficient,xalpha,numroot,alpha)
*
*           Generate various plots (EDIT the file to remove comments for specific
*           options)
*

            call ALPHAPLOT(Ncomp,j,molefrac,alpha,coefficient,Ki)

*           ALPHAROOT has internal output section to compile a table
*           listing statistics on the determination of alpha

 2000  continue

************************************************************************
*           Produce this format to plot data points as dots:
*                                  (PLOTFAT=20)
*
*                                  2
*                                  x(1)  y(1)
*                                  x(1)  y(1)
*                                  2
*                                  x(2)  y(2)
*                                  x(2)  y(2)
*                                     etc.
************************************************************************

        do 3000 j = 1, Npress

              write(8,3500)  alpha(j),xalpha(j),alpha(j),xalpha(j)
 3500         format('2'/,e16.9,10x,e16.9/,e16.9,10x,e16.9)

 3000  continue

        CLOSE(unit=1)
        CLOSE(unit=7)
        CLOSE(unit=8)

        stop
        end

************************************************************************
*
*                                    4 Dec 91
*
*                               BRETT D. WEIGLE
*                 Petroleum and Natural Gas Engineering Section
*                          Mineral Engineering Department
*                       College of Earth and Mineral Sciences
*                          The Pennsylvania State University
```

```
*                      University Park, Pennsylvania
*
*                               M.S. thesis
*
*                           SUBROUTINE ALPHACOEFF
*
*         This subroutine calculates the coefficient for each term in
*         the general polynomial for the vapor fraction, alpha:
*
*         P(alpha) = c0 + c1*alpha + c2*alpha**2 + ... +
*
*                                            c(Ncomp-1)*alpha**(Ncomp-1)
*         Equation 4.29 in the thesis.
*
************************************************************************

      SUBROUTINE ALPHACOEFF(Ncomp,Npress,jj,molefrac,Ki,coefficient)

      IMPLICIT REAL*8(a-h,o-z)
      REAL*8 Ki(500,100), molefrac(0:100)
      INTEGER p

      DIMENSION coefficient(0:100), c(100)

      OPEN(unit=14,file='coeff',status='unknown')
      OPEN(unit=15,file='coeff.plot',status='unknown')

      if(Ncomp .LT. 2) then
        write(6,*)'You cannot flash this system'
        stop
      end if

*
*         Calculate Ci = Ki - 1
*
      do 0500 k = 1, Ncomp
          c(k) = Ki(jj,k) - 1.d00
 0500 continue
*
*         p-loop increments the power of alpha
*

C       write(15,*)Ncomp

      do 1000 p = 1, Ncomp
          temporary = 0.d00

          do 2000 j = 1, p

*         Zero-order elementary symmetric function, a0[1/Ci], defined as 1
*
              if(p-j .EQ. 0) then
                apj = 1.d00
                go to 2500
```

```
              end if

*          Call subroutine to calculate the elementary symmetric
*          function, apj

              call SYMFUNCTION(Ncomp,j,p,c,apj)

 2500         ratio = 0.d00

              do 3000 i = 1, Ncomp
                  ratio = ratio + molefrac(i)/(c(i)**(j-1))
 3000         continue

              temporary = temporary + ((-1.d0)**(j+1))*apj*ratio
 2000     continue

        coefficient(Ncomp-p) = temporary

C         write(14,*)'Coefficient(',Ncomp-p,') = ',coefficient(Ncomp-p)
C         write(15,*)Ncomp-p,coefficient(Ncomp-p)

 1000 continue

        return
        end

***********************************************************************
*
*                               4 Dec 91
*
*                           BRETT D. WEIGLE
*              Petroleum and Natural Gas Engineering Section
*                      Mineral Engineering Department
*                   College of Earth and Mineral Sciences
*                     The Pennsylvania State University
*                       University Park, Pennsylvania
*
*                               M.S. thesis
*
*                         SUBROUTINE SYMFUNCTION
*
*       This subroutine calculates the elementary symmetric function
*
*                              a(p-j){1/Ci}
*
***********************************************************************

        SUBROUTINE SYMFUNCTION(Ncomp,j,p,c,apj)

        IMPLICIT REAL*8(a-h,o-z)
        REAL*8 mmatrix(100,100)
        INTEGER factor,p

        DIMENSION c(100), s(100)
```

```
*
*          Compute the power-sum series:  s = sigma[ (1/Ci)**lambda ]
*

      n = p - j
      do 1000 lambda = 1, n
          sum = 0.d00

          do 2000 i = 1, Ncomp
              sum = sum + (1.d0/c(i))**lambda
 2000     continue
          s(lambda) = sum
 1000 continue

*          Build the matrix MMATRIX

      do 3000 k = 1, n

          do 4000 l = 1, n
              if(l .LE. k) mmatrix(k,l) = s(k-l+1)
              if(l .EQ. k+1) mmatrix(k,l) = DFLOAT(k)
              if(l .GT. k+1) mmatrix(k,l) = 0.d00
 4000     continue

 3000 continue

*          Since a1{1/Ci} forms a [1x1] matrix, its determinant is the
*          element itself

      if(p-j .EQ. 1) then
         det = mmatrix(1,1)
         go to 5000
      end if
*
*          Compute the determinant of MMATRIX
*

      call DETERM(mmatrix,n,det)

*
*          Compute the elementary symmetric function
*

 5000 apj = det/factor(n)

      return
      end

***********************************************************************
*
*                    Function to compute the factorial
*
***********************************************************************
```

```
      FUNCTION factor(n)
      INTEGER factor,i,n

      factor = 1
      if(n .GT. 0) then
        do 6000 i = 2,n
           factor = factor*i
 6000   continue
      end if
      end

**********************************************************************
*
*                              4 Dec 91
*
*                          BRETT D. WEIGLE
*              Petroleum and Natural Gas Engineering Section
*                   Mineral Engineering Department
*                College of Earth and Mineral Sciences
*                  The Pennsylvania State University
*                   University Park, Pennsylvania
*
*                             M.S. thesis
*
*                          SUBROUTINE DETERM
*
*  This program calculates the determinant of an NxN matrix.
*  First, partial pivoting is performed on a nonsingular matrix by
*  Gaussian elimination. This produces a triangular matrix whose
*  determinant can be calculated by computing the product of all
*  the diagonal entries.
*  The augmented matrix does not contain the normal last column
*  which represents the right-hand side of a system of linear
*  equations; AUG is the same as the original matrix.
*
*  VARIABLES:
*           N = dimension of matrix
*           AUG = augmented matrix
*           I,J,K = indices
*           MULT = multiplier used to eliminate an unknown
*           PIVOT = used to find nonzero diagonal entry
**********************************************************************

      SUBROUTINE DETERM(aug,n,det)

      IMPLICIT REAL*8(a-h,o-z)
      REAL*8 mult
      INTEGER pivot
      DIMENSION aug(100,100)
*
*        Gaussian elimination
*
```

```
      do 7000 i = 1, n
*
*         Locate nonzero entry
*
          if(aug(i,i) .EQ. 0) then
             pivot = 0
             j = i + 1
3000         if((pivot .EQ. 0) .AND. (j .LE. n)) then
                if(aug(j,i) .NE. 0) pivot = j
                j = j + 1
                go to 3000
             end if

             if(pivot .EQ. 0) then
                print *,'Matrix is singular'
                stop
             else
*
*         Interchange rows I and PIVOT
*
             do 4000 j = i, n
                 temp = aug(i,j)
                 aug(i,j) = aug(pivot,j)
                 aug(pivot,j) = temp
4000         continue

             end if

          end if
*
*         Eliminate I-th unknown from equations I+1, ..., N
*
          do 6000 j = i+1, n
              mult = -aug(j,i) / aug(i,i)

              do 5000 k = i, n
                  aug(j,k) = aug(j,k) + mult * aug(i,k)
5000          continue

6000      continue

7000  continue

*         Calculate the determinant of matrix AUG by computing the
*         product of the diagonal elements

      prod = 1.d0
      do 8000 i = 1, n
```

```
            do 9000 j = 1, n
               if(i .EQ. j) prod = prod * aug(i,j)
 9000       continue

 8000 continue

         det = prod

         return
         end

**********************************************************************
*
*                              4 Dec 91
*
*                           BRETT D. WEIGLE
*               Petroleum and Natural Gas Engineering Section
*                    Mineral Engineering Department
*                  College of Earth and Mineral Sciences
*                    The Pennsylvania State University
*                    University Park, Pennsylvania
*
*                              M.S. thesis
*
*                          SUBROUTINE ALPHAROOT
*
*        Subroutine uses an interval-halving technique to find
*        the best root value to initialize the Newton-Raphson (N-R)
*        iterative calculations which determine the real root of
*        the alpha polynomial on the interval [0,1].
*
*        PARAMETERS: delta = alpha increment
*                    epsilon = alpha convergence criterion
*        VARIABLES:  alower = lower bound of alpha increment
*                     aupper = upper bound of alpha increment
*                     falpha = the alpha polynomial
*                     fprime = first derivative of alpha polynomial
*                     guess = iterative variable for alpha
*                     guess0 = initial estimate for N-R
*                     intcount = # of intervals until sign change
*                     iter = # of iterations until N-R converged
*                     isign,isign2 = flags for function sign change
*                     isign,isign2 = flags for function sign change
*                     numroot = flag for # of zeros (from BUDAN)
**********************************************************************

         SUBROUTINE ALPHAROOT(j,Ncomp,coefficient,xalpha,numroot,
       @                            alpha)

         IMPLICIT REAL*8(a-h,o-z)
         INTEGER p

         DIMENSION alpha(500), coefficient(0:100), xalpha(500)
```

```
      PARAMETER(delta = 0.01d0, epsilon = 1.d-06)


*
*          Write table heading
*


      write(7,*)'The Fourier-Budan Theorem yields ',numroot,' roots on
     @this interval'
      write(7,3500)
 3500 format('Intervals',4x,'Initial Guess',4x,'Iterations',4x,'Calc.
     @Alpha',4x,'Exp. Alpha')

*          Check flag NUMROOT provided by subroutine BUDAN to determine
*          root-search scheme

      if(numroot .EQ. 0) then
         write(6,*) 'No root on the interval [0,1] for data set ',j
         intcount = 0
         write(7,3900) intcount,xalpha(j)
 3900    format(i4,61x,f5.3)
         write(7,*)'No root on the interval [0,1]'
         return
      end if

      if(numroot .EQ. 1) then
         ilower = 0
         iupper = 0
      end if

      if(numroot .GE. 2) then
         ilower = 0
         iupper = 1
      end if

*          Use incremental search to determine initial guess
*                    Interval Endpoint DO-Loop

      do 0400 jroot = ilower, iupper

          intcount = 0
*
*          Test the polynomial at endpoint for initial sign value
*


          if(jroot. EQ. ilower) then
             guess = DFLOAT(ilower)
             alower = guess
             aupper = alower + delta
          end if

          if(jroot. EQ. iupper) then
             guess = DFLOAT(iupper)
```

```
             aupper = guess
             alower = aupper - delta
          end if
          ichange = 0

 0600     falpha = 0.d0
          do 1500 p = 1, Ncomp

               term = coefficient(Ncomp-p)*guess**(Ncomp-p)
               if( (Ncomp-p) .EQ. 0 ) term = coefficient(0)
               falpha = falpha + term

 1500     continue
*
*         Initialize ISIGN2 on first pass with endpoint
*

          if(ichange .EQ. 0) then
             if(falpha .GE. 0.) then
                isign2 = 1
             else
                isign2 = 0
             end if
          end if

*
*         Note the sign of the function
*

          if(falpha .GE. 0.) then
             isign = 1
          else
             isign = 0
          end if

*         Test function for sign change and increment or decrement the
*         search variable as appropriate

          if(isign2 .EQ. isign) then
             if(jroot .EQ. ilower) then
                alower = aupper
                aupper = aupper + delta
                guess = aupper
             else if(jroot .EQ. iupper) then
                aupper = alower
                alower = aupper - delta
                guess = alower
             end if
          end if

*
*         Exit subroutine if no sign change is detected on interval [0,1]
*
```

```
            if( (guess .GT. 1.) .OR. (guess .LT. 0.) ) then
               write(6,*) 'No root on the interval [0,1]'
               write(7,3800) intcount,xalpha(j)
3800          format(i4,61x,f5.3)
               write(7,*)'No root on the interval [0,1]'
               return
            end if
*
*           If NO sign change but still within interval, repeat the sequence
*
            if(isign .EQ. isign2) then
               isign2 = isign
               intcount = intcount + 1
               ichange = 1
               go to 0600

            else

*           If there IS a sign change:
*           Halve the interval where the function crosses the x axis

               guess0 = (alower + aupper) / 2.d0
            end if
*
*           Provide this guess to Newton-Raphson to begin calculations
*

            guess = guess0

*
*           N-R is limited to 1000 iterations for convergence
*

            iter = 0

            do 1000 iterlimit = 1, 1000

                iter = iter + 1
                falpha = 0.d00
                fprime = 0.d00

                do 2000 p = 1, Ncomp
                    falpha = falpha + coefficient(Ncomp-p)
     @                          *guess**(Ncomp-p)
                    fprime = fprime + (Ncomp-p)*coefficient(Ncomp-p)
     @                          *guess**(Ncomp-p-1)
 2000          continue

                calc = guess - falpha/fprime
                error = DABS((calc - guess)/calc)
                guess = calc
                if(error .LE. epsilon) go to 3000
```

```
 1000      continue

      print *,'N-R method failed to converge after 1000 iterations'

*
*          Output results to file "TABLE"
*

 3000      write(7,3600) intcount,guess0,iter,guess,xalpha(j)
 3600      format(i4,13x,f5.3,10x,i4,13x,f9.6,7x,f5.3)

           alpha(j) = guess
*
*          Begin search for root from opposite end of interval
*

 0400 continue

      return
      end

**************************************************************************
*
*                              6 Dec 91
*
*                           BRETT D. WEIGLE
*             Petroleum and Natural Gas Engineering Section
*                    Mineral Engineering Department
*                  College of Earth and Mineral Sciences
*                   The Pennsylvania State University
*                     University Park, Pennsylvania
*
*                              M.S. thesis
*
*                        SUBROUTINE ALPHAPLOT
*
*          This subroutine is used for several purposes:
*          1. Plotting F(alpha) vs alpha [Rachford-Rice obj function]
*          2. Plotting F(alpha) vs alpha [polynomial]
*          3. Plotting Fprime vs alpha [polynomial]
*
**************************************************************************

      SUBROUTINE ALPHAPLOT(Ncomp,J,molefrac,alpha,coefficient,Ki)

      IMPLICIT REAL*8 (a-h,o-z)
      REAL*8 Ki(500,100),molefrac(100)
      DIMENSION alpha(500),coefficient(0:100)

      INTEGER p

      PARAMETER(start = 0.0d0, end = 2.0d0, stepsize = 0.0005d0)

      OPEN(unit=11,file='fa.plot',status='unknown')
```

```
      OPEN(unit=12,file='fprime.plot',status='unknown')

*
*        Number of data points for plotting
*

      number = IDINT((end - start + stepsize)/stepsize)

************************************************************************
*           F(alpha) vs alpha [polynomial]
*           F'(alpha) vs alpha [polynomial]
*                  Adjust Ncomp,Npress in PARAMETER

      write(11,*) number
      do 1000 phase = start,end,stepsize
         falpha = 0.d00
         fprime = 0.d00
         do 2000 p = 1,Ncomp
            falpha = falpha + coefficient(Ncomp-p)*
     @                   phase**(Ncomp-p)
C           fprime = fprime + (Ncomp-p)*coefficient(Ncomp-p)*
C    @                    phase**(Ncomp-p-1)
 2000    continue
         write(11,3600) phase,falpha
C        write(11,3600) phase,fprime
 3600    format(f7.3,2x,f25.12)
 1000 continue
************************************************************************
C*                     Rachford-Rice objective function
C
C       do 4500 k = 1,Npress
C       k = 6
C       write(11,*) number
C          do 3000 phase = start,end,stepsize
C             falpha = 0.d00
C             do 4000 i = 1,Ncomp
C                falpha = falpha + (molefrac(i)*(Ki(k,i) - 1.d0)) /
C    @                        (1.d00 + phase*(Ki(k,i) - 1.d0))
C*                     End of i loop
C 4000        continue
C
C             write(11,3500) phase,falpha
C 3500        format(f7.3,2x,f25.12)
C*                     End of phase loop
C 3000     continue
C*                     End of k loop
C 4500 continue
************************************************************************

      CLOSE(unit=11)
      CLOSE(unit=12)

      return
      end
```

```
*********************************************************************
*
*                              6 Dec 91
*
*                           BRETT D. WEIGLE
*               Petroleum and Natural Gas Engineering Section
*                     Mineral Engineering Department
*                  College of Earth and Mineral Sciences
*                    The Pennsylvania State University
*                     University Park, Pennsylvania
*
*                              M.S. thesis
*
*                          SUBROUTINE BUDAN
*
*       Subroutine uses the Fourier-Budan Theorem to determine
*       the number of roots that the alpha polynomial has on the
*      interval [u,v].
*
*       PARAMETERS: iu = lower bound of alpha interval
*                   iv = uppper bound of alpha interval
*       VARIABLES:  coefficient = coefficient of alpha polynomial
*                   dcoeff = coefficient of polynomial derivatives
*                   deriv = derivatives of alpha polynomial
*                   fvapor = the alpha polynomial
*                   ia,ib = # of sign changes for derivative series
*                   ivapor = alpha = vapor fraction
*                   jsign,ksign = flags for derivative sign change
*                   numroot = number of zeros on the interval
*********************************************************************

      SUBROUTINE BUDAN(J,Ncomp,coefficient,numroot)

      IMPLICIT REAL*8(a-h,o-z)
      INTEGER p

      DIMENSION dcoeff(0:100,0:100), coefficient(0:100), deriv(0:100)

      PARAMETER(iu = 0, iv = 1)


C       DATA (coefficient(l), l = 0,Ncomp-1) /-1.,1.,-2.,3.,-4.,5. /
      OPEN(unit=2,file='test',status='unknown')
      REWIND(unit=2)

      ia = 0
      ib = 0
      do 0500 ivapor = iu, iv, 1
*
*        Evaluate the polynomial function at the endpoints iu and iv
*

         fvapor = 0.d0
```

```
         do 0600 p = 1, Ncomp
            fvapor = fvapor + coefficient(Ncomp-p)*ivapor**(Ncomp-p)
 0600   continue
               write(2,*) 'fvapor = ',fvapor
               write(2,*) ' '

*
*        Calculate coefficients of first derivative
*

         do 1000 n = Ncomp-1, 0, -1
            dcoeff(0,n) = coefficient(n)
            write (2,*) 'dcoeff(0,',n,') = ',dcoeff(0,n)
 1000   continue
               write(2,*) ' '

*        Calculate coefficients of 2nd- and higher-order derivatives
*        as multiples of those of the first derivative

         do 1500 m = 1, Ncomp-1

            do 2000 n = Ncomp-m, 1, -1
               dcoeff(m,n-1) = n*dcoeff(m-1,n)
            write (2,*) 'dcoeff(',m,',',n-1,') = ',
     @dcoeff(m,n-1)
 2000      continue
               write(2,*) ' '

 1500   continue

*
*        Evaluate the derivative series at the endpoints iu and iv
*

         do 3000 m = 1, Ncomp-1
            deriv(m) = 0.d0

            do 4000 n = Ncomp-m, 1, -1
               term = dcoeff(m,n-1)*ivapor**(n-1)
               if( (n-1) .EQ. 0 ) term = dcoeff(m,n-1)
               deriv(m) = deriv(m) + term
               write(2,*) 'inter deriv(',m,') = ',deriv(m)
 4000      continue

               write(2,*) 'total deriv(',m,') = ',deriv(m)
               write(2,*) ' '

 3000   continue

*
*        Count the sign changes between the terms of the series
*

         if(fvapor. LT. 0.) then
```

```
           ksign = 0
         else
           ksign = 1
         end if
               write(2,*) 'ksign = ',ksign,' for fvapor'

         do 5000 i = 1, Ncomp-1
            if(deriv(i) .LT. 0.) then
              jsign = 0
            else
              jsign = 1
            end if
               write(2,*) 'jsign = ',jsign,' for deriv(',i,')'

*
*        Increment A or B, depending upon the endpoint under evaluation
*

            if(ivapor .EQ. iu) then
              if(ksign .NE. jsign) then
                ia = ia + 1
                write(2,*) 'ia = ',ia,' for deriv(',i,')'
              end if
            end if

            if(ivapor .EQ. iv) then
              if(ksign .NE. jsign) then
                ib = ib + 1
                write(2,*) 'ib = ',ib,' for deriv(',i,')'
              end if
            end if

            ksign = jsign
              write(2,*) 'ksign = ',ksign,' after deriv(',i,')'
              write(2,*) ' '
 5000   continue

 0500 continue

*
*        Pass a flag to calling program to indicate root conditions
*

      write(2,*) 'ia = ',ia,' and ib = ',ib
      numroot = ia - ib
      write(2,*) 'numroot = ',numroot

      write(2,6000) Ncomp-1, numroot, iu, iv, J
 6000 format('This polynomial of order ',i3,' has ',i3,' zeros on the in
     @terval [',i2,',',i2,'] for J = ',i3)

      CLOSE(unit=2)
      return
      end
```

The Pennsylvania State University

The Graduate School

Department of Mineral Engineering

A GENERALIZED POLYNOMIAL FORM OF THE OBJECTIVE FUNCTION IN FLASH CALCULATIONS

A Thesis in

Petroleum and Natural Gas Engineering

by

Brett D. Weigle

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 1992

I grant The Pennsylvania State University the nonexclusive right to use this work for the University's own purposes and to make single copies of the work available to the public on a not-for-profit basis if copies are not otherwise available.

Brett D. Weigle

Brett D. Weigle

We approve the thesis of Brett D. Weigle.

Date of Signature

Michael Adewumi
Michael A. Adewumi
Associate Professor of Petroleum
and Natural Gas Engineering
Thesis Advisor

12/18/91

Turgay Ertekin
Professor of Petroleum and
Natural Gas Engineering
Section Chair of Petroleum and
Natural Gas Engineering

12/18/91

Paul J. Hicks
Paul J. Hicks, Jr.
Assistant Professor of Petroleum
and Natural Gas Engineering

12-18-91

David M. Bressoud
David M. Bressoud
Professor of Mathematics

12/18/91

ABSTRACT

The Rachford-Rice objective function for flash calculations exhibits a nearly flat slope across the two-phase region and sharp discontinuities near the dewpoint. These features make iterative solution procedures sensitive to the initial estimate of the root and prone to spurious values if a correction step throws the algorithm outside the two-phase region or near the phase boundary.

This work centers on the recasting of the Rachford-Rice objective function into a polynomial function of the vapor fraction, α. The degree of this polynomial is one less than the number of components in the system and its coefficients can be calculated from the feed composition and the equilibrium ratios. A recursive expression is developed that involves symmetric functions and can be easily programmed on a computer or scientific calculator.

The principal advantage of this new form of the objective function is that the theory of polynomials is well-developed. The location of their zeros can be predicted with confidence by techniques based on sound mathematical principles, such as the Fourier-Budan theorem. The α-polynomial is well-behaved over the two-phase region and its root can be quickly located by a hybrid method of interval-halving technique and Newton-Raphson procedure. The validity of the new objective function and its automatic coefficient-generating algorithm are tested using several multicomponent systems for which experimental data are available.

Results of these tests prove conclusively the validity of the generalized polynomial objective function. The versatility of this form of the flash objective function, compared with the original Rachford-Rice version, is demonstrated. Another

potential advantage of the polynomial form is its ability to handle dilute systems in which some components are present but in very low concentrations. It also promises possible usage as a means of developing appropriate lumping schemes.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

Roman

A = pressure-dependent constant [Equation (2.6)]

A = parameter for the PREOS [Equation (2.18)]

A = number of sign changes in derivative series (Section 5.2.4)

a = intercept of log Kp vs F plot [Equation (2.3)]

a = unitary or elementary symmetric function [Equation (3.41)]

$a(T)$ = attractive constant for PREOS [Equation (2.11)]

$a(T_c)$ = attractive constant at critical point for PREOS [Equation (2.10)]

B = parameter for the PREOS [Equation (2.19)]

B = number of sign changes in derivative series (Section 5.2.4)

b = translation constant for Cox chart [Equation (2.1)]

b = molar co-volume for PREOS [Equation (2.9)]

C_i = constant used in the objective function [Equation (3.16)]

$\hat{C}_i$ = excluded term [Equation (3.42)]

c = slope of log Kp vs F plot [Equation (2.3)]

c = volume translation parameter [Equation (2.21)]

F = number of moles in feed stream (Section 3.1.1)

F = component characterization factor [Equation (2.1)]

K_i = equilibrium ratio (Section 1.2)

L = number of moles in the liquid phase (Section 3.1.1)

$\mathbf{M}$ = interexpressibility matrix [Equation (4.12)]

m = element of matrix $\mathbf{M}$ [Equation (4.31)]

N = number of components in the fluid system (Section 1.3)

n = number of terms in the set [Equation (3.41)]

P = a partition of r into at most λ parts [Equation (4.8)]

p = pressure

p_i = part of a partition [Equation (4.1)]

$p(t)$ = polynomial as a function of t (Section 5.2.4)

$p(u)$ = polynomial as a function of u (Section 5.2.4)

$p(v)$ = polynomial as a function of v (Section 5.2.4)

R = real gas constant [Equation (2.8)]

$\mathbf{R}$ = field of real numbers (Section 5.2.4)

s = power sum symmetric function [Equation (4.6)]

s_P = slope of plot [Equation (2.2)]

T = temperature

t = exponential term in generating function [Equation (3.41)]

u = augmented unitary symmetric function [Equation (4.7)]

u = real number (Section 5.2.4)

V = number of moles in the vapor phase (Section 3.1.1)

v = molar volume

$\tilde{v}$ = pseudo volume [Equation (2.21)]

v = length of a partition [Equation (4.2)]

v = real number (Section 5.2.4)

w = weight of a partition [Equation (4.1)]

X = fluid "map" coordinate from Varotsis (Section 2.2)

X_i = component "map" coordinate from Varotsis (Section 2.2)

x = mole fraction in the liquid phase (Section 3.1.1)

x = argument of symmetric function [Equation (3.41)]

Y = fluid "map" coordinate from Varotsis (Section 2.2)

Y_i = component "map" coordinate from Varotsis (Section 2.2)

y = mole fraction in the vapor phase (Section 3.1.1)

Z = compressibility factor [Equations (2.20), (2.22)]

z = mole fraction in the feed stream (Section 3.1.1)

Greek

α = vapor fraction [Equation (1.1)]

β = isothermal compressibility (Section 2.3.2)

δ = binary interaction coefficient [Equation (2.16)]

δ = bulk modulus [Equation (2.23)]

η = function to describe $a(T)$ away from critical point [Equation (2.11)]

κ = function of acentric factor in PREOS [Equation (2.13)]

κ_0 = function of acentric factor in PRSV EOS [Equation (2.26)]

κ_1 = parameter in PRSV EOS [Equation (2.27)]

μ = coefficient of polynomial [Equation (5.2)]

π_i = multiplicity of a part in a partition [Equation (4.1)]

ψ = EOS variable [Equations (2.24), (2.25)]]

ω_i = Pitzer acentric factor for the i-th component [Equation (2.5)]

Subscripts

B = boiling point [Equation (2.1)]

c = critical property

i,j,k,l,m = individual components of the fluid system

ij = interaction between component i and component j of the fluid system

k = convergence pressure (Section 2.2)

P = constant pressure [Equation (4.29)]

R = reduced property

RA = Rackett compressibility factor [Equation (2.22)]

r = order of symmetric function [Equation (3.41)]

T = constant temperature [Section 2.3.2, Equation (4.29)]

Superscripts

I = fluid state [Equations (2.24), (2.25)]

i = component of fluid system [Equations (2.24), (2.25)]

(n) = order of derivative

$'$ = first derivative

$''$ = second derivative

$'''$ = third derivative

Abbreviations

CPU = computer central processing unit

EOS = equation of state

°F = degree Fahrenheit

LHS = left-hand side (of an equation)

PREOS = Peng-Robinson equation of state

PRSV EOS = Peng-Robinson-Stryjek-Vera equation of state

psia = pounds per square inch, absolute

Q.E.D. = *quod erat demonstrandum*, which was to be proved (Appendix A)

RHS = right-hand side (of an equation)

SRKEOS = Soave-Redlich-Kwong equation of state

SSM = successive substitution method

ACKNOWLEDGMENTS

The author acknowledges the United States Army for affording him the opportunity to study for this degree.

The author expresses gratitude to his thesis advisor, Dr. Michael A. Adewumi, whose enthusiasm was a driving force behind this research. His keen grasp of the physics of phase behavior and his mathematical expertise were invaluable in seeing it to its completion.

The author wishes to thank Captain John H. Warren, United States Army, for his pioneering work in this field. Without the foundation provided by his thesis, this work would not have reached conception, much less fruition.

The author expresses appreciation to Dr. David M. Bressoud for his lessons in combinatory analysis and symmetric functions. He provided the clue that finally yielded the recursive form of the α-polynomial.

The author offers his appreciation to the other members of his thesis committee, Dr. Turgay Ertekin and Dr. Paul J. Hicks, Jr., for their time and comments in evaluating this work.

Mr. Timothy Kohler is the source of most of the author's ability to efficiently utilize the department's computers and was always there to answer a question or deal with a recalcitrant machine.

The author extends his gratitude to his wife, Deborah Lee Weigle, for cheerfully tolerating a work schedule worse than the one imposed on the author as a company commander. Her support and understanding were instrumental in the completion of this thesis.

Chapter 1

DISCUSSION OF THE PROBLEM

1.1 Introduction

Determination of the equilibrium state of coexisting liquid and vapor phases, particularly for multicomponent fluid mixtures, is of vital interest to the petroleum and chemical industries. Many processes in petroleum production and refining involve repetitive flash calculations for design and operational purposes. The primary goal of performing flash calculations is to determine the relative amounts and compositions of the coexisting phases for a given feed composition at a specified condition of temperature and pressure.

This work is confined solely to two-phase vapor-liquid equilibrium computations, although its results will no doubt find application in multiphase flash problems in the future.

1.2 The Generic Flash Algorithm

To begin the calculation, the following variables must be specified: the system pressure and temperature, the molar composition of the feed stream, z_i , and an initial estimate of the equilibrium ratios, $K_i = \frac{y_i}{x_i}$. The process is assumed to occur under isothermal and isobaric conditions. The stages of the calculation are:

1. Compute initial estimates of the equilibrium ratios by one of the established techniques or by an empirical correlation.

2. Calculate the phase distribution and compositions corresponding to the given K-values. This involves the iterative solution of the following objective function, developed by Rachford and Rice (1952):

$$\sum_{i=1}^{N} \frac{z_i \, (K_i - 1)}{1 + \alpha \, (K_i - 1)} = 0 \tag{1.1}$$

   where α is the vapor molar fraction.

3. Use an equation of state (EOS) to calculate the component fugacities in each phase and check for equality.

4. If equality is not achieved (i.e., the phases are not in equilibrium), correct the K-values on the basis of the fugacities and repeat steps 2-4.

This correction is commonly performed using a successive substitution-type method or a second-order Newton-type scheme. These algorithms are well-known and are described in several papers [e.g., Risnes et al. (1981); Michelsen (1982); Boston and Britt (1978)].

Successful implementation of the generic flash algorithm described above requires three principal elements. These are (1) a general estimate of the set of equilibrium ratios to start the procedure; (2) a good equation of state to improve K_i; and (3) a robust objective function that guarantees convergence to a single value of α. A poor first guess of K-values may produce a phase split that is physically impossible under the prevailing pressure and temperature. Satisfactory methods are available for generating these values. Furthermore, existing equations of state do a fairly good job of predicting phase properties, and other efforts continue along this line.

One area that has not enjoyed equal amounts of attention for a long time is the form of the objective function. Invariably, the Rachford-Rice objective function [Equation (1.1)] is most often used. Recent investigations (Warren, 1991) have shown that this objective function does exhibit some strange behavior which may affect its ability to generate good results for some conditions.

1.3 Objectives of the Investigation

This study is aimed at evolving a generalization of the new polynomial form of the Rachford-Rice objective function developed by Warren (1991). The resulting generalized polynomial function of the vapor fraction, α, should be capable of handling an N-component mixture. The coefficients of the generalized polynomial should depend on only two variables, the molar composition of the feed stream and the equilibrium ratios, and should be easy to obtain, either analytically or numerically. Appropriate algorithms are to be developed for this purpose.

The principal advantage of a polynomial form of the flash-calculation objective function is that the theory of polynomials is well-developed and semi-analytical solution techniques exist for equations up to fifth-order (Zaguskin, 1961). For higher-order polynomials, the Newton-Raphson iterative method usually provides a fast and accurate determination of the roots.

Determination of all the zeros of this polynomial is unnecessary since the physics of the problem demands that only the zeros on the bounded interval [0,1] are of practical interest. Furthermore, the physics also suggests that only one zero (or value of α) exists on this interval, which represents the two-phase vapor-liquid region. It can be shown mathematically that this is indeed the case for well-defined systems, as will be demonstrated in 5.2.

Chapter 2

LITERATURE REVIEW

A survey of the pertinent literature reveals that apparently only one other worker, Warren (1991), has studied the particular aspect of flash calculations targeted in this research. A comprehensive review of the literature pertaining to the use of cubic equations of state in flash calculations was conducted in order to provide a reference point for the testing of the polynomial objective function.

This review is sub-divided into three sections: flash calculation algorithms; equilibrium ratios; and cubic equations of state. Particular emphasis is laid on the Peng-Robinson equation of state.

2.1 Vapor-Liquid Equilibrium Flash Calculations

This discussion will be confined to two-phase vapor-liquid equilibria. The work to date concentrates on developing robust algorithms with rapid convergence rates. Robustness implies the ability to continue the calculations after recovering from a spurious value of the vapor fraction computed in the neighborhood of the critical point or at the phase boundaries. Abhvani and Beaumont (1987) present an excellent review of EOS-based flash algorithms. They divide the papers into two categories according to solution method, those using some variant of the successive substitution method (SSM) or those employing a second-order Newton-type method.

The SSM technique is the traditional solution algorithm, but it exhibits a poor rate of convergence and [illegible] stability problems close to saturation points and in the critical region. Risnes et al. (1981), Michelsen (1982), and Mehra et al. (1983) made attempts at acceleration and stabilization of this method.

Similarly, many workers have proposed various forms of second-order Newton procedures to avoid the slow rate of convergence of the SSM, such as Boston and Britt (1978), Fussell and Yanosik (1978), Asselineau et al. (1979), Fussell (1979), Baker and Luks (1980), and Varotsis et al. (1981). Others advocate a combination of successive-substitution and Newton methods; the former is used to provide good initial values to the rapidly converging latter. Informative studies include Mott (1980, 1983), Mehra et al. (1982), Michelsen (1982), Nghiem et al. (1983), and Abhvani and Beaumont (1987).

Benmekki (1984) developed a general algorithm for flash calculations that can utilize any cubic equation of state and features a specified calculational path for computing the phase boundaries. This is an attempt to ensure that bubblepoint and dewpoint computations originate from within the two-phase region, thus guaranteeing meaningful values of the equilibrium ratios.

Warren (1991) made a radical departure from previous efforts at enhancing flash calculation algorithms when he formulated an explicit linear equation for the vapor fraction of a binary system. He successfully extended this to a quadratic equation for a ternary system and a cubic equation for a quaternary mixture. The success achieved by Warren and the possibility of the existence of a generalized polynomial expression for the vapor fraction in a two-phase, N-component fluid system motivated the current work.

2.2 Vapor-Liquid Equilibrium Ratios

The use of initial equilibrium ratios close to the final values for a multicomponent fluid is crucial to the rapid convergence of any flash calculation. Experimental values are preferred because the prediction of K_i for a particular fluid at various combinations of temperature, pressure and composition requires lengthy calculations. Therefore, predictive methods for K-values are a limiting factor in the speed and robustness of any flash calculation algorithm.

The expression "equilibrium constant" was coined by Souders et al. (1932) and was defined as the ratio of the vapor mole fraction to that of the liquid. The basis for most predictive methods had its genesis when Cox (1923) observed that the lines on a semilogarithmic plot of vapor pressure against temperature appeared to converge to a single pressure. Katz and Hachmuth (1937) demonstrated an analogous behavior for equilibrium constants; they converged to unity at a fluid mixture's critical pressure.

White and Brown (1942) attempted to develop a correlation for K-values based on this "convergence" pressure. Hanson and Brown (1945) used experimental data to correlate the convergence pressure (p_k) at one temperature as a function of the molal average boiling point of the equilibrium vapor and liquid. They also showed that the convergence pressure concept could be extended from binary to multicomponent systems.

Hadden (1948, 1953) produced nomographs for equilibrium constants of pure components as functions of temperature and pressure, and incorporated convergence and vapor pressures into nomographs for mixtures. He demonstrated that mixture convergence pressure is a function of the operating temperature and of the liquid-phase composition exclusive of the lightest component concentration. This composition dependence led Muskat (1949) to propose the use of the term "equilibrium ratio" in place of "equilibrium constant." Edmister (1949) presented a graph involving the ratio of differences between the convergence and critical pressures and the ratio of differences between the system and critical temperatures.

Winn (1952) developed nomographs based on Hadden's (1948) results that allow the determination of K-values at a convergence pressure of 5000 psia. For systems with $p_k \neq 5000$, he provides a translation to find the value of K_i at other "apparent" convergence pressures. The methods proposed by these three authors require charts and do not lend themselves to computer calculations.

Hoffmann et al. (1953) attempted to extend Cox's (1923) vapor pressure graph for the purpose of determining equilibrium ratios by plotting log Kp against the component characterization factor F, where

Kp = product of equilibrium ratio and pressure

$$F = b\left[\frac{1}{T_B} - \frac{1}{T}\right] \tag{2.1}$$

b = constant required to translate the vapor pressure curve for a hydrocarbon onto the straight line of the Cox chart
T_B = hydrocarbon boiling point
T = system temperature

Brinkman and Sicking (1960) presented an iterative method for finding the convergence pressure based on the slope, s_P, of the plot mentioned in Hoffmann et al. (1953). Then, the equilibrium ratio could be determined as

$$K = \frac{p_k}{p}\, e^{s_P F} \tag{2.2}$$

Standing (1979) observed that the composition dependence of the equilibrium ratio is negligible at pressures below 1000 psia. He proceeded to combine the work of Hoffmann et al. (1953) and Brinkman and Sicking (1960) to develop a correlation for K-values for the crude oils studied by Katz and Hachmuth (1937):

$$K = \frac{1}{p} 10^{(a + cF)} \tag{2.3}$$

where a and c are the intercept and slope (respectively) of log Kp vs. F plots of the abovementioned oils. Both a and c are expressed as functions of pressure. He also presented equations for the heavy fraction and the common reservoir gases N_2, CO_2 and H_2S (referred to as permanent gases).

Wilson (1969) published a K-value equation that currently enjoys widespread use in flash calculations:

$$K_i = \frac{e^B}{p_{Ri}} \tag{2.4}$$

where

$$B = 5.37(1 + \omega_i)(1 - \frac{1}{T_{Ri}}) \tag{2.5}$$

p_{Ri} = reduced pressure of the i-th component
T_{Ri} = reduced temperature of the i-th component
ω_i = Pitzer acentric factor of the i-th component

Wilson's equation fails to predict accurate equilibrium ratios for most fluids above pressures of 500 psia, as illustrated by Warren (1991). Whitson and Torp (1981) attempted to correct this problem by re-introducing the system convergence pressure to the Wilson equation:

$$K_i = \left[\frac{p_{ci}}{p_k} \right]^{A - 1} \frac{e^{AB}}{p_{Ri}} \tag{2.6}$$

where

$$A = 1 - \left[\frac{p - 14.7}{p_k - 14.7} \right]^{0.6} \tag{2.7}$$

p_{ci} = critical pressure of the i-th component

Risnes and Dalen (1984) took an approach based on the equation of state used in the flash calculations. Their basic idea was to assume the mixture or feed to be liquid and then evaporate up to one-half of the system to form a gas phase by use of the fugacities. The initial K-values then could be calculated from the resulting phases. This method is reported to perform well near the critical point and along the bubblepoint line but often fails along the dewpoint curve.

Reportedly, the most accurate K-value predictor is that proposed by Varotsis (1989). He used over 1000 experimental equilibrium ratios to construct an X-Y plot such that each reservoir fluid's position on the "map" is determined by its coordinates X and Y. These coordinates are described by a polynomial fitted to the apparent pressure mentioned in Winn (1952). He proposes an equation for the convergence pressure based on the mole fraction of methane and nitrogen in the fluid.

Each pure hydrocarbon component is represented on the map by its own set of coordinates (X_i, Y_i), which are calculated as functions of the component acentric factor. Specific values are given for the permanent gases and correlations based on molecular weight are specified for the heptane-plus fraction.

Finally, the straight line that joins the pressure and temperature coordinates (X, Y) of the fluid with the position of each component on the map (X_i, Y_i) intersects the K-value axis at a point that corresponds to the equilibrium value of the selected constituent. Varotsis (1989) presents tables for three different crude oils and gas condensates at varying temperatures and pressures that display K-values remarkably close to experimental values. They are an order-of-magnitude improvement over those predicted by the equations of Wilson (1969) or Whitson and Torp (1981).

The method of Varotsis (1989) was attempted in the current work. His K-value predictor was formulated using data for crude oils and gas condensates containing the $C_1 - C_6$ alkane series, the heptane-plus pseudocomponent and the permanent gases. It will not properly describe systems (such as the methane-ethane-propane ternary) containing fewer components than these "typical" reservoir fluids. For lack of a suitable replacement expression for p_k , Wilson's equation is used in the current work.

2.3 Cubic Equations of State (EOS)

The equation of state (EOS) is the heart of a modern flash calculation algorithm. Ideally, it should be able to accurately represent the thermodynamic properties of the fluid of interest over the complete range of operating pressures and temperatures. Since engineering applications rarely focus on an isolated chemical species, the EOS should incorporate mixing rules that allow it to extend its predictive capabilities to the behavior of multicomponent fluids. Its component-specific descriptive parameters should be readily calculable from well-known properties, such as critical temperature and pressure, molecular weight and acentric factor. Finally, the associated computations should not consume excessive computer time, especially if the equation of state is to be used for repetitive calculations.

The engineer is faced with the choice of using a complex EOS exhibiting a high degree of non-linearity and many adjustable parameters, or a cubic EOS which possesses an analytical solution and requires the estimation of two or three parameters. Mathias and Benson (1986) presented a comparison of average central-processing-unit (CPU) times required by three cubic EOS and by three complicated EOS to compute fugacity coefficients and enthalpy departures. They asserted that the time required for any of the candidate equations to calculate the density root (or compressibility factor root) is negligible compared to that involved in executing the various mixing rules. In fact, for systems containing more than about six components, the cubic EOS become more computationally burdensome than the complicated EOS simply because of the cross terms inherent to the cubic EOS mixing rules.

Engineers frequently use cubic EOS because they work well over the range of most industrial operating conditions and are easily programmed for solution on a computer. The two cubic EOS which have gained the widest acceptance are Soave's modifications of the Redlich-Kwong (1949) equation of state (SRKEOS) (Soave, 1972) and that presented by Peng and Robinson (1976b) (PREOS). The PREOS and suggested improvements are examined in this work for possible use in flash calculations because of the author's familiarity with this EOS.

2.3.1 Development of the Peng-Robinson EOS

Upon the success of the SRKEOS, Peng et al. (1975) undertook a further study to formulate a cubic equation of state with an improved capability to predict liquid densities and other fluid properties, particularly in the vicinity of the critical region.

This study resulted in a further modification of the attractive pressure term of the classical equation of state proposed by van der Waals (1873). The result was the EOS presented by Peng and Robinson (1976b):

$$p = \frac{RT}{v-b} - \frac{a(T)}{v(v+b)+b(v-b)} \tag{2.8}$$

The values of the parameters are obtained from

$$b = 0.07780\,\frac{RT_c}{p_c} \tag{2.9}$$

$$a(T_c) = 0.45724\,\frac{R^2 T_c^{\,2}}{p_c} \tag{2.10}$$

$$a(T) = a(T_c)\cdot\eta(T_R,\,\omega) \tag{2.11}$$

$$\eta^{1/2} = 1 + \kappa\,(1 - T_R^{\,1/2}) \tag{2.12}$$

$$\kappa = 0.3746 + 1.4850\omega - 0.1644\omega^2 + 0.01667\omega^3 \tag{2.13}$$

Equation (2.12) has the same form as that used by Soave (1972) but κ was obtained by fitting a larger range of vapor pressure data as a function of the reduced temperature and the acentric factor (Pitzer et al., 1955) of each component.

In order to use the equation for systems containing more than one component, the following mixing rules are presented:

$$a = \sum_i \sum_j x_i x_j a_{ij} \tag{2.14}$$

$$b = \sum_i x_i b_i \tag{2.15}$$

where

$$a_{ij} = (1 - \delta_{ij})a_i^{\,1/2} a_j^{\,1/2} \tag{2.16}$$

Equations (2.14) and (2.15) are consequences of the mixing rule proposed by Kay (1936), while Equation (2.16) was developed by Zudkevitch and Joffe (1970). The experimentally determined binary interaction coefficient, δ_{ij}, characterizes the binary formed by component i and component j. The importance of δ_{ij} in accurately reproducing P-V-T data was discussed by Peng and Robinson (1976b) and by Robinson et al. (1985).

The PREOS can be written in the form of a cubic equation in the compressibility factor:

$$Z^3 - (1 - B)Z^2 + (A - 3B^2 - 2B)Z - (AB - B^2 - B^3) = 0 \tag{2.17}$$

where

$$A = \frac{ap}{R^2T^2} \tag{2.18}$$

$$B = \frac{bp}{RT} \tag{2.19}$$

$$Z = \frac{pv}{RT} \tag{2.20}$$

2.3.2 Selection of the Proper Root in Cubic EOS

Equation (2.17) yields one or three roots depending upon the number of phases in the system. The authors stated that, in the two-phase region, the largest root is for the compressibility factor of the vapor while the smallest positive root corresponds to that of the liquid.

Lawal (1987), however, asserted that this criterion was insufficient to select the proper root. He proved that, in the event of multiple real roots, the smallest of the positive roots *larger than or equal to B* must be chosen for the compressibility of the liquid.

Asselineau et al. (1979) compared the calculated volume to the pseudo-critical volume to assign the root to the proper phase, under specific conditions. Poling et al. (1981) examined the order of magnitude of the isothermal compressibility, $\beta = -(\partial v/\partial p)_T/v$, to ascertain the presence of the liquid or vapor phase. Gosset et al. (1986) offered two discriminants, one based on the Cardan criterion for the number of real roots for a cubic equation and a heuristic approach similar to that of Asselineau et al. (1979).

2.3.3 Modifications to the Peng-Robinson EOS

Numerous attempts have been made to correct for the deficiencies inherent in a cubic equation of state by introducing additional parameters into the PREOS. These changes improve some aspect of the EOS's performance (usually liquid density predictions) but at the cost of increased complexity and the requirement for tables or correlations to determine the additional parameter(s) for each fluid component. This review will touch on a limited number of these studies.

2.3.3.1 Volume Corrections

The modification of the SRKEOS proposed by Péneloux et al. (1982) also formed the basis for two other studies concerned with the PREOS. These authors suggested that the use of a "pseudo volume" defined by

$$\tilde{v} = v + \sum_i c_i x_i \tag{2.21}$$

could be used to effect a translation along the volume axis, leaving unchanged the predicted equilibrium conditions. They chose c so that correct saturated liquid densities were exactly reproduced at the reduced temperature $T_R = 0.7$. They rejected the acentric factor as a correlating parameter in favor of the Rackett compressibility factor, Z_{RA}, developed by Spencer and Danner (1972):

$$c = 0.40768 \frac{RT_c}{p_c}(0.29441 - Z_{RA}) \tag{2.22}$$

Their third parameter did improve predictions of saturated liquid densities.

Almost simultaneously, Jhaveri and Youngren (1988) and Mathias et al. (1989) presented three-parameter modifications of the PREOS based on the work of Péneloux et al. (1982). The first authors correlated the third parameter, c, with molecular weight. The second study retained the Péneloux-Rauzy-Fréze volume correction scheme but added a *further* term involving the bulk modulus to handle the critical region. The bulk modulus is dimensionless and is defined as:

$$\delta = -\frac{v^2}{RT}\left[\frac{\partial p}{\partial v}\right]_T \tag{2.23}$$

From an examination of the graphs accompanying both publications, the work of Mathias et al. (1989) seems to produce results closer to the experimental values for saturated volumes and densities.

2.3.3.2 Temperature Dependence

Xu and Sandler (1987a,b) postulated that the molar co-volume term, b, is not independent of temperature and they disputed the fitting of vapor pressures used by Peng and Robinson (1976b) to characterize the attractive constant, a. They correlated vapor pressure and volume data for 16 components at both subcritical and supercritical conditions and proposed to replace the numeric coefficients of a and b found in Equation (2.9) and Equation (2.10) with:

$$\psi_a{}^{\mathrm{I}} = \sum_{i=0}^{2} a_i{}^{\mathrm{I}} T_R^i \qquad (2.24)$$

and

$$\psi_b{}^{\mathrm{I}} = \sum_{i=0}^{2} b_i{}^{\mathrm{I}} T_R^i \qquad (2.25)$$

where i refers to the species and I denotes either subcritical or supercritical conditions.

Wu and Sandler (1989) generalized the temperature-dependent parameters of Xu and Sandler (1987a,b) by performing least-squares fits of ψ_a and ψ_b as functions of acentric factor and reduced temperature. They were able to accomplish this task only for the n-alkane series because of insufficient data. For their intended application of the work (petroleum reservoir simulation), they envisioned the use of the fluid-specific parameters for the permanent gases, water and light ends and the generalized parameters for the heavy pseudocomponents.

Stryjek and Vera (1986a,b,c,d) re-worked Equation (2.13) to obtain a better reproduction of vapor pressure data at low reduced temperatures:

$$\kappa_0 = 0.378893 + 1.4897153\omega - 0.17131848\omega^2 + 0.0196554\omega^3 \qquad (2.26)$$

and modified Equation (2.12) by the introduction of one compound-characteristic adjustable parameter, κ_1:

$$\kappa = \kappa_0 + \kappa_1\,(1 + T_R{}^{1/2})(0.7 - T_R) \qquad (2.27)$$

Stryjek and Vera (1986b) and Proust and Vera (1989) listed values of κ_1 for over 160 compounds of industrial interest. Stryjek and Vera (1986d) and Wilczek-Vera and Vera (1987) examined mixing rules of varying complexity for use with the Peng-Robinson-Stryjek-Vera (PRSV) EOS. For the current work, the PRSV EOS with the original PREOS mixing rules (as formulated by Stryjek and Vera, 1986b) is used and produces noticeably better results than did the PREOS.

2.3.4 References on Cubic EOS

Abbott (1979) and Martin (1979) presented comprehensive reviews of cubic equations of state available at that time, and Vidal (1983) and Vera et al. (1984) updated the topic. Huron and Vidal (1979) proposed composition-dependent mixing rules while Mathias and Copeman (1983) discussed mixing rules dependent on volume. Finally, Peng and Robinson (1976b), Peng and Robinson (1977), Robinson and Peng (1978), Robinson (1979) and Peng (1986) developed specific applications of their EOS.

Chapter 3

DEVELOPMENT OF THE POLYNOMIAL FUNCTION FOR SIMPLE SYSTEMS

This chapter discusses the work published by Rachford and Rice (1952) and Warren (1991) on performing flash calculations. It shows the development of the Rachford-Rice objective function [Equation (1.1)] and extends Warren's work as a precursor to developing a generalized, multicomponent equation for the vapor fraction.

3.1 The Rachford-Rice Flash Objective Function

We will briefly examine the derivation of the Rachford-Rice objective function that is universally used today in flash calculations. After plotting its behavior, it will become plain why it is so difficult to solve by iterative techniques such as the Newton-Raphson method.

3.1.1 The Material Balance Development

Flash calculations are used to determine the compositions and quantities of the vapor and liquid phases at equilibrium which result when an N-component fluid of a particular composition is subjected to a particular pressure and temperature. The composition of the feed stream, F, is denoted by Σz_i and it flashes into L moles of liquid with composition Σx_i, and V moles of vapor with composition Σy_i. The resulting material balance equations are:

$$F = L + V \tag{3.1}$$

$$Fz_i = Lx_i + Vy_i \tag{3.2}$$

As defined in Chapter 1, the equilibrium ratio is:

$$K_i = \frac{y_i}{x_i} \tag{3.3}$$

and, rearranging, one is left with the equation:

$$y_i = x_i K_i \tag{3.4}$$

Substituting Equation (3.4) into Equation (3.2) yields:

$$Fz_i = Vx_i K_i + Lx_i \tag{3.5}$$

Simplify by isolating the x_i term and dividing through by F:

$$z_i = x_i \left[\frac{VK_i}{F} + \frac{L}{F} \right] \tag{3.6}$$

Dividing Equation (3.1) through by F and solving for $\frac{L}{F}$ yields:

$$\frac{L}{F} = 1 - \frac{V}{F} \tag{3.7}$$

Substituting Equation (3.7) in Equation (3.6) and simplifying the equation results in:

$$x_i = \frac{z_i}{1 + \frac{V}{F} (K_i - 1)} \tag{3.8}$$

Imposing the constraint of $\sum_{i=1}^{N} x_i = 1$ on Equation (3.8) leaves:

$$1 = \sum_{i=1}^{N} \frac{z_i}{1 + \frac{V}{F} (K_i - 1)} \tag{3.9}$$

Rearranging:

$$\sum_{i=1}^{N} \left[\frac{z_i}{1 + \frac{V}{F} (K_i - 1)} \right] - 1 = 0 \tag{3.10}$$

Recalling Equation (3.4), we can write:

$$y_i = \frac{z_i K_i}{1 + \frac{V}{F} (K_i - 1)} \tag{3.11}$$

Imposing the constraint of $\sum_{i=1}^{N} y_i = 1$ on Equation (3.11) yields:

$$\sum_{i=1}^{N} \left[\frac{z_i K_i}{1 + \frac{V}{F} (K_i - 1)} \right] - 1 = 0 \tag{3.12}$$

Combining Equation (3.10) and Equation (3.12) leaves:

$$\sum_{i=1}^{N} \frac{z_i \, (K_i - 1)}{1 + \frac{V}{F} \, (K_i - 1)} = 0 \qquad (3.13)$$

Defining the vapor fraction, α, as:

$$\alpha = \frac{V}{F} \qquad (3.14)$$

and substituting Equation (3.14) into Equation (3.13) yield the Rachford-Rice objective function:

$$\sum_{i=1}^{N} \frac{z_i \, (K_i - 1)}{1 + \alpha \, (K_i - 1)} = 0 \qquad (3.15)$$

3.1.2 A Graphic Representation of the Rachford-Rice Objective Function

As the authors noted, their formulation of the objective function was prone to instability near the values of α that represented the phase boundaries, namely, 0 and 1. They showed that the slope of the function near these points may be quite steep. It is this feature that tends to throw derivative-based root-finding techniques out of the two-phase region, yielding spurious roots.

Figure 3.1 depicts the behavior of the objective function over a wide range of α for a binary system of 70% methane and 30% ethane (Bloomer et al., 1953). Figure 3.2 does the same for a ternary system consisting of 85% methane, 10% ethane and 5% propane (Parikh et al., 1984). Although values of the vapor fraction

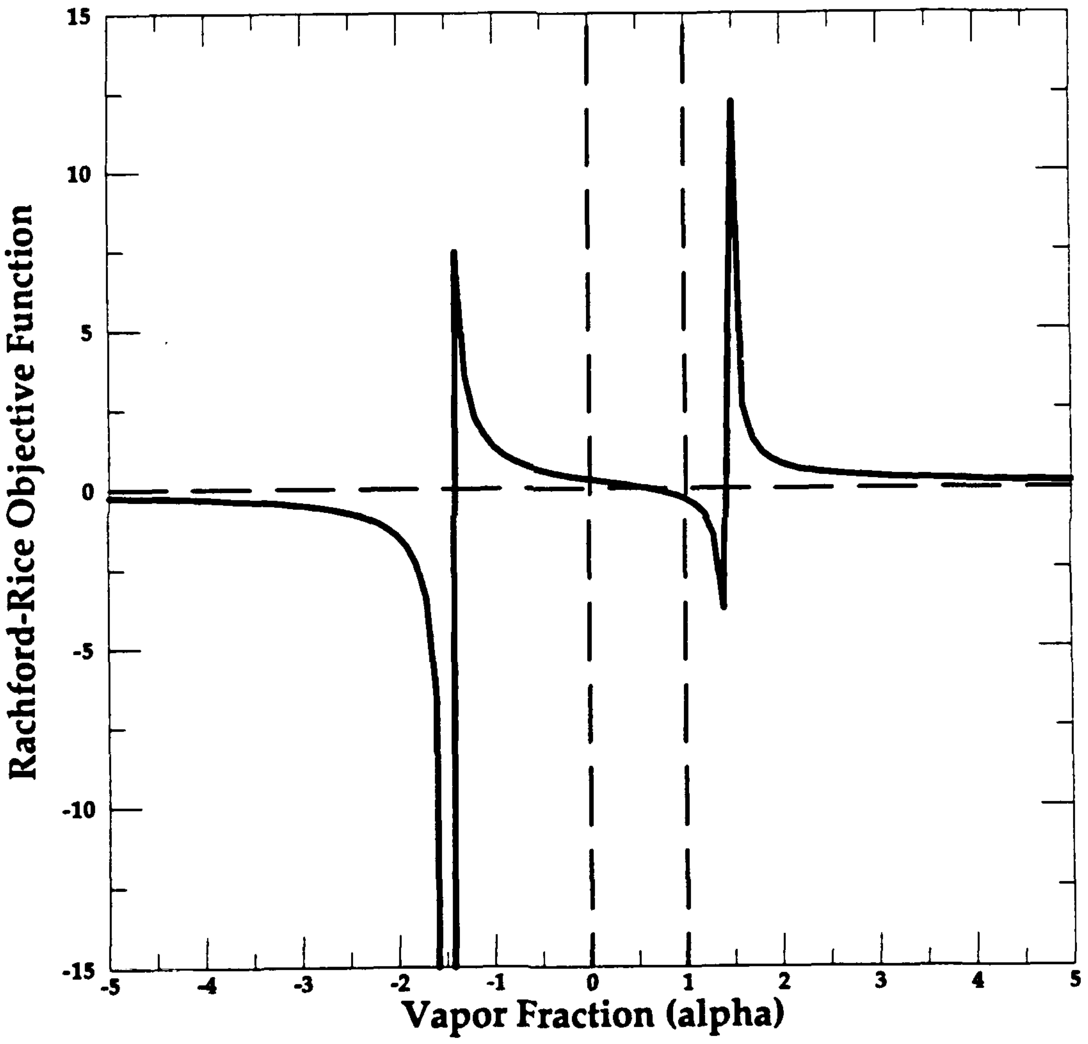


Figure 3.1: The Behavior of the Rachford-Rice Objective Function for a Methane-Ethane Binary System

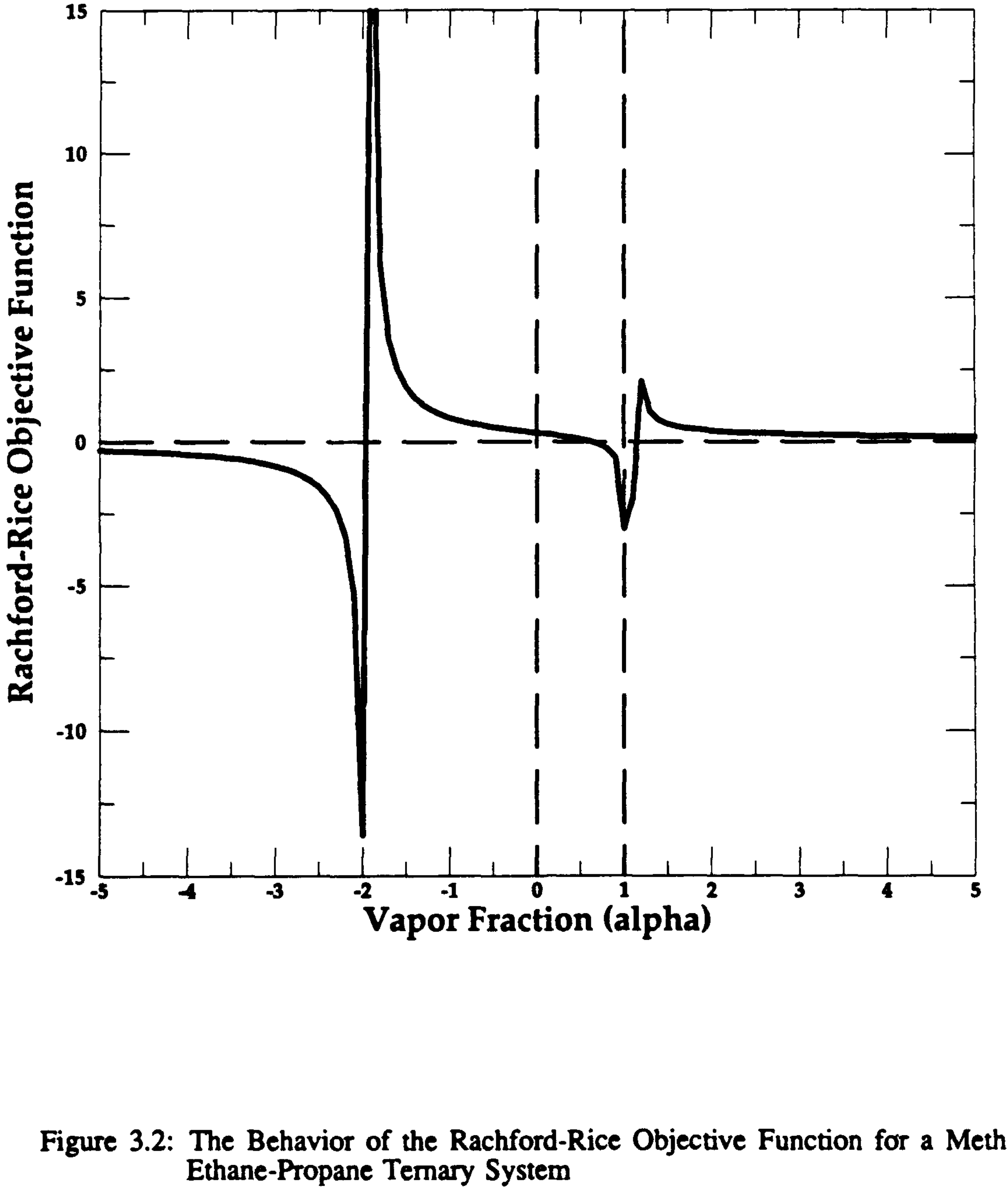


Figure 3.2: The Behavior of the Rachford-Rice Objective Function for a Methane-Ethane-Propane Ternary System

have no physical meaning outside the interval [0,1], these graphs serve to illustrate how ill-behaved the objective function is. Its slope is almost flat as it traverses the two-phase region and it is plagued by spiky singularities.

This work will attempt to develop a new expression for α, one that possesses reasonable slope over the desired interval and has no discontinuities near the phase boundaries.

3.2 Warren's Explicit Equations for the Vapor Fraction

Warren (1991) expanded the Rachford-Rice objective function into a polynomial in α for a binary, ternary and quaternary fluid system. He did this by setting N equal to 2, 3 or 4, respectively, and reducing the resulting equations to their simplest polynomial form by algebraic manipulations. To demonstrate the validity of his work, Warren also showed that the higher-order polynomials would reduce to those for smaller systems when the appropriate mole fractions and equilibrium constants were removed.

We will assume (as did Warren) that, under isobaric and isothermal conditions, the equilibrium constant does not change such that the quantity $(K_i - 1)$, which appears in the objective function, can be represented by a constant, C_i.

We will reproduce the entire process here for a binary system but will show only the final result for a ternary and quaternary system, since the algebra can be quite tedious and repetitive.

3.2.1 Binary System

Starting with the objective function as defined in Equation (3.15):

$$\sum_{i=1}^{N} \frac{z_i \, (K_i - 1)}{1 + \alpha \, (K_i - 1)} = 0 \tag{3.15}$$

and defining $C_i = K_i - 1$, Equation (3.15) can be rewritten as:

$$\sum_{i=1}^{N} \frac{z_i \, C_i}{1 + \alpha \, C_i} = 0 \tag{3.16}$$

For a *two-component* system, setting $N = 2$ in Equation (3.16) and expanding term-wise yields:

$$\frac{z_1 \, C_1}{1 + \alpha \, C_1} + \frac{z_2 \, C_2}{1 + \alpha \, C_2} = 0 \tag{3.17}$$

Moving the terms with the subscript "2" to the right-hand side of the equation:

$$\frac{z_1 \, C_1}{1 + \alpha \, C_1} = - \frac{z_2 \, C_2}{1 + \alpha \, C_2} \tag{3.18}$$

By multiplying each side by $(1 + \alpha \, C_1) \, (1 + \alpha \, C_2)$, one obtains:

$$(z_1 \, C_1) \, (1 + \alpha \, C_2) = - \, (z_2 \, C_2) \, (1 + \alpha \, C_1) \tag{3.19}$$

Expanding each side yields:

$$z_1 \, C_1 + \alpha \, z_1 \, C_1 \, C_2 = - \, z_2 \, C_2 - \alpha \, z_2 \, C_1 \, C_2 \tag{3.20}$$

Move α terms to the left-hand side of the equation and all remaining terms to the right-hand side, then recall that, for a binary system, $z_1 + z_2 = 1$:

$$\alpha\,(C_1\,C_2) = -\,(z_1\,C_1 + z_2\,C_2) \tag{3.21}$$

Dividing both sides through by $C_1\,C_2$ and substituting $(K_i - 1)$ yields the explicit form of the objective function for a binary system:

$$\alpha = -\left[\frac{z_1}{K_2 - 1} + \frac{z_2}{K_1 - 1}\right] \tag{3.22}$$

3.2.2 Ternary System

$$\alpha^2 + \alpha\left[\frac{\sum\limits_{i=1}^{3} z_i \left[\sum\limits_{j\neq i}^{3} C_j\right]}{\prod\limits_{j\neq i}^{3} C_j}\right] + \sum_{i=1}^{3}\frac{z_i}{\prod\limits_{j\neq i}^{3} C_j} = 0 \tag{3.23}$$

3.2.3 Quaternary System

$$\alpha^3 + \alpha^2\left[\sum_{i=1}^{4}\frac{(1 - z_i)}{C_i}\right] + \alpha\left[\frac{\sum\limits_{i=1}^{4} z_i \left[\sum\limits_{j\neq i}^{4} C_j\right]}{\prod\limits_{j\neq i}^{4} C_j}\right] + \sum_{i=1}^{4}\frac{z_i}{\prod\limits_{j\neq i}^{4} C_j} = 0 \tag{3.24}$$

3.3 Extension of Warren's Work to Larger Systems

Warren's method can be used to develop polynomial expressions for systems having five, six and seven components. It will be observed that the terms of the equation expand in a regular fashion, thereby suggesting the possibility of developing a recursive relationship dependent only on N, the number of components. Only the algebra for the five-component system will be presented, as those for six- and seven-component systems follow the same procedure.

3.3.1 Quinary System

We begin with the objective function:

$$\sum_{i=1}^{5} \frac{z_i\, C_i}{1 + \alpha\, C_i} = 0 \tag{3.16}$$

which, when expanded for five components, becomes:

$$\frac{z_1\, C_1}{1 + \alpha\, C_1} + \frac{z_2\, C_2}{1 + \alpha\, C_2} + \frac{z_3\, C_3}{1 + \alpha\, C_3} + \frac{z_4\, C_4}{1 + \alpha\, C_4} + \frac{z_5\, C_5}{1 + \alpha\, C_5} = 0 \tag{3.25}$$

Multiplying through by $\prod_{i=1}^{5} (1 + \alpha\, C_i)$ yields:

$$\begin{aligned}
&z_1\, C_1 (1 + \alpha\, C_2)(1 + \alpha\, C_3)(1 + \alpha\, C_4)(1 + \alpha\, C_5) + \\
&z_2\, C_2 (1 + \alpha\, C_1)(1 + \alpha\, C_3)(1 + \alpha\, C_4)(1 + \alpha\, C_5) + \\
&z_3\, C_3 (1 + \alpha\, C_1)(1 + \alpha\, C_2)(1 + \alpha\, C_4)(1 + \alpha\, C_5) + \\
&z_4\, C_4 (1 + \alpha\, C_1)(1 + \alpha\, C_2)(1 + \alpha\, C_3)(1 + \alpha\, C_5) + \\
&z_5\, C_5 (1 + \alpha\, C_1)(1 + \alpha\, C_2)(1 + \alpha\, C_3)(1 + \alpha\, C_4) = 0
\end{aligned} \tag{3.26}$$

Expanding each term:

$$\alpha^4\, z_1\, C_1\, C_2\, C_3\, C_4\, C_5 +$$

$$
\begin{aligned}
&\alpha^3 z_1 C_1 (C_2 C_3 C_4 + C_2 C_3 C_5 + C_2 C_4 C_5 + C_3 C_4 C_5) + \\
&\alpha^2 z_1 C_1 (C_2 C_3 + C_2 C_4 + C_2 C_5 + C_3 C_4 + C_3 C_5 + C_4 C_5) + \\
&\alpha z_1 C_1 (C_2 + C_3 + C_4 + C_5) + z_1 C_1 + \\
\\
&\alpha^4 z_2 C_1 C_2 C_3 C_4 C_5 + \\
&\alpha^3 z_2 C_2 (C_1 C_3 C_4 + C_1 C_3 C_5 + C_1 C_4 C_5 + C_3 C_4 C_5) + \\
&\alpha^2 z_2 C_2 (C_1 C_3 + C_1 C_4 + C_1 C_5 + C_3 C_4 + C_3 C_5 + C_4 C_5) + \\
&\alpha z_2 C_2 (C_1 + C_3 + C_4 + C_5) + z_2 C_2 + \\
\\
&\alpha^4 z_3 C_1 C_2 C_3 C_4 C_5 + \\
&\alpha^3 z_3 C_3 (C_1 C_2 C_4 + C_1 C_2 C_5 + C_1 C_4 C_5 + C_2 C_4 C_5) + \\
&\alpha^2 z_3 C_3 (C_1 C_2 + C_1 C_4 + C_1 C_5 + C_2 C_4 + C_2 C_5 + C_4 C_5) + \\
&\alpha z_3 C_3 (C_1 + C_2 + C_4 + C_5) + z_3 C_3 + \\
\\
&\alpha^4 z_4 C_1 C_2 C_3 C_4 C_5 + \\
&\alpha^3 z_4 C_4 (C_1 C_2 C_3 + C_1 C_2 C_5 + C_2 C_3 C_5 + C_1 C_3 C_5) + \\
&\alpha^2 z_4 C_4 (C_1 C_2 + C_1 C_3 + C_1 C_5 + C_2 C_3 + C_2 C_5 + C_3 C_5) + \\
&\alpha z_4 C_4 (C_1 + C_2 + C_3 + C_5) + z_4 C_4 + \\
\\
&\alpha^4 z_5 C_1 C_2 C_3 C_4 C_5 + \\
&\alpha^3 z_5 C_5 (C_1 C_2 C_3 + C_1 C_2 C_4 + C_1 C_3 C_4 + C_2 C_3 C_4) + \\
&\alpha^2 z_5 C_5 (C_1 C_2 + C_1 C_3 + C_1 C_4 + C_2 C_3 + C_2 C_4 + C_3 C_4) + \\
&\alpha z_5 C_5 (C_1 + C_2 + C_3 + C_4) + z_5 C_5 = 0
\end{aligned}
\tag{3.27}
$$

Dividing through by the term $\prod_{i=1}^{5} C_i$ and adding like terms yields:

$$
\alpha^4 + \alpha^3 \left\{ z_1 \left[\frac{1}{C_2} + \frac{1}{C_3} + \frac{1}{C_4} + \frac{1}{C_5} \right] + z_2 \left[\frac{1}{C_1} + \frac{1}{C_3} + \frac{1}{C_4} + \frac{1}{C_5} \right] + \right.
$$

$$
z_3 \left[\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_4} + \frac{1}{C_5} \right] + z_4 \left[\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \frac{1}{C_5} \right] +
$$

$$z_5\left[\frac{1}{C_1}+\frac{1}{C_2}+\frac{1}{C_3}+\frac{1}{C_4}\right]\Bigg]\Bigg\}+$$

$$\alpha^2\Bigg\{z_1\left[\frac{1}{C_2C_3}+\frac{1}{C_2C_4}+\frac{1}{C_2C_5}+\frac{1}{C_3C_4}+\frac{1}{C_3C_5}+\frac{1}{C_4C_5}\right]+$$

$$z_2\left[\frac{1}{C_1C_3}+\frac{1}{C_1C_4}+\frac{1}{C_1C_5}+\frac{1}{C_3C_4}+\frac{1}{C_3C_5}+\frac{1}{C_4C_5}\right]+$$

$$z_3\left[\frac{1}{C_1C_2}+\frac{1}{C_1C_4}+\frac{1}{C_1C_5}+\frac{1}{C_2C_4}+\frac{1}{C_2C_5}+\frac{1}{C_4C_5}\right]+$$

$$z_4\left[\frac{1}{C_1C_2}+\frac{1}{C_1C_3}+\frac{1}{C_1C_5}+\frac{1}{C_2C_3}+\frac{1}{C_2C_5}+\frac{1}{C_3C_5}\right]+$$

$$z_5\left[\frac{1}{C_1C_2}+\frac{1}{C_1C_3}+\frac{1}{C_1C_4}+\frac{1}{C_2C_3}+\frac{1}{C_2C_4}+\frac{1}{C_3C_4}\right]\Bigg\}+$$

$$\alpha\Bigg\{z_1\left[\frac{C_2+C_3+C_4+C_5}{C_2C_3C_4C_5}\right]+z_2\left[\frac{C_1+C_3+C_4+C_5}{C_1C_3C_4C_5}\right]+$$

$$z_3\left[\frac{C_1+C_2+C_4+C_5}{C_1C_2C_4C_5}\right]+z_4\left[\frac{C_1+C_2+C_3+C_5}{C_1C_2C_3C_5}\right]+$$

$$z_5\left[\frac{C_1+C_2+C_3+C_4}{C_1C_2C_3C_4}\right]\Bigg\}+\frac{z_1}{C_2C_3C_4C_5}+\frac{z_2}{C_1C_3C_4C_5}+$$

$$\frac{z_3}{C_1C_2C_4C_5}+\frac{z_4}{C_1C_2C_3C_5}+\frac{z_5}{C_1C_2C_3C_4}=0 \qquad (3.28)$$

To maintain similarity with the forms of the quaternary and ternary equations, we can separate the general term in the coefficient for α in Equation (3.28) into four fractions:

$$z_i\left[\frac{C_j+C_k+C_l+C_m}{C_j\,C_k\,C_l\,C_m}\right]=z_i\left[\frac{1}{C_k\,C_l\,C_m}+\frac{1}{C_j\,C_l\,C_m}+\frac{1}{C_j\,C_k\,C_m}+\frac{1}{C_j\,C_k\,C_l}\right] \qquad (3.29)$$

After multiplying each of the five fractions by z_i and collecting terms with common denominators, the following form appears:

$$\frac{z_i + z_j}{C_k\, C_l\, C_m} \tag{3.30}$$

We can invoke $\sum_{i=1}^{5} z_i = 1$ to construct fractions with similar subscripted terms in both numerator and denominator:

$$\frac{1 - z_k - z_l - z_m}{C_k\, C_l\, C_m} \tag{3.31}$$

This yields a final polynomial expression of the same general form as those of Warren (1991):

$$\begin{aligned}
&\alpha^4 + \alpha^3 \left\{ \frac{1 - z_1}{C_1} + \frac{1 - z_2}{C_2} + \frac{1 - z_3}{C_3} + \frac{1 - z_4}{C_4} + \frac{1 - z_5}{C_5} \right\} + \\
&\alpha^2 \left\{ \frac{1 - z_1 - z_2}{C_1\, C_2} + \frac{1 - z_1 - z_3}{C_1\, C_3} + \frac{1 - z_1 - z_4}{C_1\, C_4} + \frac{1 - z_1 - z_5}{C_1\, C_5} + \frac{1 - z_2 - z_3}{C_2\, C_3} + \right. \\
&\left. \frac{1 - z_2 - z_4}{C_2\, C_4} + \frac{1 - z_2 - z_5}{C_2\, C_5} + \frac{1 - z_3 - z_4}{C_3\, C_4} + \frac{1 - z_3 - z_5}{C_3\, C_5} + \frac{1 - z_4 - z_5}{C_4\, C_5} \right\} + \\
&\alpha \left\{ \frac{1 - z_1 - z_2 - z_3}{C_1 C_2 C_3} + \frac{1 - z_1 - z_2 - z_4}{C_1 C_2 C_4} + \frac{1 - z_1 - z_2 - z_5}{C_1 C_2 C_5} + \frac{1 - z_1 - z_3 - z_4}{C_1 C_3 C_4} \right. \\
&+ \frac{1 - z_1 - z_3 - z_5}{C_1 C_3 C_5} + \frac{1 - z_1 - z_4 - z_5}{C_1 C_4 C_5} + \frac{1 - z_2 - z_3 - z_4}{C_2 C_3 C_4} + \frac{1 - z_2 - z_3 - z_5}{C_2 C_3 C_5} \\
&\left. + \frac{1 - z_2 - z_4 - z_5}{C_2 C_4 C_5} + \frac{1 - z_3 - z_4 - z_5}{C_3 C_4 C_5} \right\} + \frac{z_1}{C_2\, C_3\, C_4\, C_5} + \frac{z_2}{C_1\, C_3\, C_4\, C_5} + \\
&\frac{z_3}{C_1\, C_2\, C_4\, C_5} + \frac{z_4}{C_1\, C_2\, C_3\, C_5} + \frac{z_5}{C_1\, C_2\, C_3\, C_4} = 0
\end{aligned} \tag{3.32}$$

This yields the polynomial expression for a quinary system:

$$\alpha^4 + \alpha^3 \left[\sum_{i=1}^{5} \frac{(1 - z_i)}{C_i} \right] + \alpha^2 \left[\sum_{i=1}^{4} \sum_{j=i+1}^{5} \frac{(1 - z_i - z_j)}{C_i\, C_j} \right] +$$

$$\alpha \left[\sum_{i=1}^{4} \sum_{j=i+1}^{5} \sum_{k=i+2}^{6} \frac{(1 - z_i - z_j - z_k)}{C_i\, C_j\, C_k} \right] + \sum_{i=1}^{5} \frac{z_i}{\prod\limits_{j \neq i}^{5} C_j} = 0 \tag{3.33}$$

3.3.2 Reduction to Quaternary System

Before proceeding to develop the equations for six- and seven-component systems, we must ensure that the quinary equation will reduce to that of a quaternary system under the proper conditions. This is accomplished by setting z_5 equal to zero and K_5 equal to one (Warren, 1991).

When z_5 becomes zero, so must x_5 and y_5. This would seem to leave K_5 undefined:

$$\lim_{\substack{x_5 \to 0 \\ y_5 \to 0}} K_5 = \frac{0}{0} = \text{undefined} \tag{3.34}$$

We can remove this difficulty by the application of l'Hospital's Rule. The expression becomes:

$$\lim_{\substack{x_5 \to 0 \\ y_5 \to 0}} K_5 = \frac{\dfrac{dy_5}{dy_5}}{\dfrac{dx_5}{dx_5}} = \frac{1}{1} = 1 \tag{3.35}$$

Therefore, $C_5 = K_5 - 1 = 1 - 1 = 0$.

To avoid division by zero, multiply Equation (3.32) by $\prod_{i=1}^{5} C_i$:

$$\begin{aligned}
&\alpha^4 C_1 C_2 C_3 C_4 C_5 + \\
&\alpha^3 \Big[(1 - z_1) C_2 C_3 C_4 C_5 + (1 - z_2) C_1 C_3 C_4 C_5 + (1 - z_3) C_1 C_2 C_4 C_5 + \\
&(1 - z_4) C_1 C_2 C_3 C_5 + (1 - z_5) C_1 C_2 C_3 C_4 \Big] + \\
&\alpha^2 \Big[(1 - z_1 - z_2) C_3 C_4 C_5 + (1 - z_1 - z_3) C_2 C_4 C_5 + (1 - z_1 - z_4) C_2 C_3 C_5 \\
&+ (1 - z_1 - z_5) C_2 C_3 C_4 + (1 - z_2 - z_3) C_1 C_4 C_5 + (1 - z_2 - z_4) C_1 C_3 C_5 + \\
&(1 - z_2 - z_5) C_1 C_3 C_4 + (1 - z_3 - z_4) C_1 C_2 C_5 + (1 - z_3 - z_5) C_1 C_2 C_4 + \\
&\qquad (1 - z_4 - z_5) C_1 C_2 C_3 \Big] + \\
&\alpha \Big[z_1 C_1 (C_2 + C_3 + C_4 + C_5) + z_2 C_2 (C_1 + C_3 + C_4 + C_5) + \\
&z_3 C_3 (C_1 + C_2 + C_4 + C_5) + z_4 C_4 (C_1 + C_2 + C_3 + C_5) + \\
&z_5 C_5 (C_1 + C_2 + C_3 + C_4) \Big] + z_1 C_1 + z_2 C_2 + z_3 C_3 + z_4 C_4 + z_5 C_5 = 0
\end{aligned} \tag{3.36}$$

Let z_5 and C_5 equal zero:

$$\begin{aligned}
&\alpha^3 C_1 C_2 C_3 C_4 + \\
&\alpha^2 \Big[(1 - z_1) C_2 C_3 C_4 + (1 - z_2) C_1 C_3 C_4 + (1 - z_3) C_1 C_2 C_4 + \\
&\qquad (1 - z_4) C_1 C_2 C_3 \Big] + \\
&\alpha \Big[z_1 C_1 (C_2 + C_3 + C_4) + z_2 C_2 (C_1 + C_3 + C_4) + z_3 C_3 (C_1 + C_2 + C_4) + \\
&\qquad z_4 C_4 (C_1 + C_2 + C_3) + z_1 C_1 + z_2 C_2 + z_3 C_3 + z_4 C_4 = 0
\end{aligned} \tag{3.37}$$

This is identical to Equation (3.21) in Warren's work, which is the expanded form of the quaternary equation. We can now safely derive the expressions for six- and seven-component systems.

3.3.3 Senary System

$$\alpha^5 + \alpha^4 \left[\sum_{i=1}^{6} \frac{(1 - z_i)}{C_i} \right] + \alpha^3 \left[\sum_{i=1}^{5} \sum_{j=i+1}^{6} \frac{(1 - z_i - z_j)}{C_i\, C_j} \right] +$$

$$\alpha^2 \left[\sum_{i=1}^{4} \sum_{j=i+1}^{5} \sum_{k=i+2}^{6} \frac{(1 - z_i - z_j - z_k)}{C_i\, C_j\, C_k} \right] +$$

$$\alpha \left[\sum_{i=1}^{3} \sum_{j=i+1}^{4} \sum_{k=i+2}^{5} \sum_{l=i+3}^{6} \frac{(1 - z_i - z_j - z_k - z_l)}{C_i\, C_j\, C_k\, C_l} \right] + \sum_{i=1}^{6} \frac{z_i}{\prod\limits_{j \neq i}^{6} C_j} = 0 \tag{3.38}$$

3.3.4 Septenary System

$$\alpha^6 + \alpha^5 \left[\sum_{i=1}^{7} \frac{(1 - z_i)}{C_i} \right] + \alpha^4 \left[\sum_{i=1}^{6} \sum_{j=i+1}^{7} \frac{(1 - z_i - z_j)}{C_i\, C_j} \right] +$$

$$\alpha^3 \left[\sum_{i=1}^{5} \sum_{j=i+1}^{6} \sum_{k=i+2}^{7} \frac{(1 - z_i - z_j - z_k)}{C_i\, C_j\, C_k} \right] +$$

$$\alpha^2 \left[\sum_{i=1}^{4} \sum_{j=i+1}^{5} \sum_{k=i+2}^{6} \sum_{l=i+3}^{7} \frac{(1 - z_i - z_j - z_k - z_l)}{C_i\, C_j\, C_k\, C_l} \right] +$$

$$\alpha \left[\sum_{i=1}^{3} \sum_{j=i+1}^{4} \sum_{k=i+2}^{5} \sum_{l=i+3}^{6} \sum_{m=i+4}^{7} \frac{(1 - z_i - z_j - z_k - z_l - z_m)}{C_i\, C_j\, C_k\, C_l - C_m} \right]$$

$$+ \sum_{i=1}^{7} \frac{z_i}{\prod\limits_{j \neq i}^{7} C_j} = 0 \tag{3.39}$$

3.4 Formulation of a Generalized Equation for the Vapor Fraction

The objective function can be recast in the following form:

$$\sum_{i=1}^{N} z_i \, C_i \prod_{j \neq i}^{N} (1 + \alpha \, C_j) = 0, \quad \text{where } C_i, C_j = \text{constant} \tag{3.40}$$

Equation (3.40) is in the form of the generating function for the *elementary symmetric functions*, a_r:

$$\prod_{i=1}^{n} (1 + t \, x_i) = \sum_{r=0}^{n} t^r \, a_r(x_i) \tag{3.41}$$

According to Macdonald (1979), $a_0(x_i) = 1$ and $a_r(x_i) = 0$ for all $r > n$.

We can now express the objective function in terms of the r-th elementary symmetric function in C_i :

$$\sum_{i=1}^{N} z_i \, C_i \prod_{j \neq i}^{N} (1 + \alpha \, C_j) = \sum_{i=1}^{N} z_i \, C_i \sum_{r=0}^{N-1} \alpha^r \, a_r(C_1, \ldots, \hat{C}_i, \ldots, C_N) = 0 \tag{3.42}$$

where $\hat{C}_i$ indicates the exclusion of the i-th term from the operation.

Since α^r does not involve i, we can invert the order of the summations:

$$\sum_{r=0}^{N-1} \alpha^r \left\{ \sum_{i=1}^{N} z_i \, C_i \, a_r(C_1, \ldots, \hat{C}_i, \ldots, C_N) \right\} = 0 \tag{3.43}$$

A working definition of the elementary symmetric function a_r could be "taking permutations of the elements of a set r terms at a time." For example,

$$a_1(C_1, C_2, \ldots, C_N) = (C_1 + C_2 + \cdots + C_N) \tag{3.44}$$

$$a_2(C_1^{-1}, C_2^{-1}, \ldots, C_N^{-1}) = \left[\frac{1}{C_1 C_2} + \frac{1}{C_1 C_3} + \cdots + \frac{1}{C_{N-1}\, C_N}\right] \qquad (3.45)$$

The condition $\hat{C}_i$ is equivalent to the $j \neq i$ condition imposed on the summation terms in the earlier versions of the α-polynomial and herein lies the computational awkwardness. We want to find an expression that allows the summation to proceed over *all* N components, which is an operation readily represented by a DO-loop in computer programming.

To eliminate $\hat{C}_i$, we must expand the symmetric function. In Chapter 4, we will tackle this problem after a discussion of symmetric functions.

Chapter 4

DEVELOPMENT OF THE GENERALIZED EQUATION

In this chapter, we shall present a brief introduction to the theory of symmetric functions to show why they provide such a powerful tool to express permutations. Then we will show the reasoning used in the search for a recursive expression for α in terms of N, C_i and z_i. Finally, we will present a generalized multicomponent equation for the vapor fraction, α, that is compact and readily programmed on a computer.

4.1 Introduction to Symmetric Functions

4.1.1 Notation and Definitions of Partitions

Any collection of v non-negative integers (excluding zero) whose sum is w is called a *v-partition* of w. The individual integers are referred to as *parts* of the partition and are conventionally written in descending order of magnitude.

David et al. (1966) state that if there are λ distinct parts, say $p_1, p_2, \ldots, p_\lambda$ with $p_1 > p_2 > p_3 > \cdots > p_\lambda \geq 1$ and if p_i is repeated π_i times, with $i = 1, 2, \ldots, \lambda$, then the partition is written $(p_1^{\pi_1} p_2^{\pi_2} \ldots p_\lambda^{\pi_\lambda})$. The *weight*, w, of the partition is written as

$$w = \sum_{i=1}^{\lambda} p_i \pi_i \tag{4.1}$$

and the number of parts, or *length*, is

$$\nu = \sum_{i=1}^{\lambda} \pi_i \tag{4.2}$$

Macdonald (1979) refers to π_i as the *multiplicity* of i in the partition. For example, the partition $(4^2 21^3)$ has weight 13, 6 parts and 3 distinct parts. In our notation,

$$p_1 = 4 \text{ and } \pi_1 = 2; \; p_2 = 2 \text{ and } \pi_2 = 1; \; p_3 = 1 \text{ and } \pi_3 = 3$$

4.1.2 Symmetric Functions

A *symmetric* function is one in which the individual parts can be interchanged without altering the value of the function, such as

$$\sum_{i=1}^{n} x_i = x_1 + x_2 + x_3 + \cdots + x_n \tag{4.3}$$

The number, n, of the quantities x does not affect the relationships between the various forms of the symmetric functions, but does appear in the final expressions. David et al. (1966) write

$$\sum_{i=1}^{n} x_i = (1), \quad \sum_{i=1}^{n} x_i^r = (r) \text{ and } \sum_{i \neq j}^{n} x_i^r x_j^s = (rs), \text{ for } r \neq s \tag{4.4}$$

This leads directly to the definitions of two special forms of symmetric functions. MacMahon (1920) defines the *unitary* or a-functions as

$$a_r = (\,1^r\,) = \sum_{i_1 < \ldots < i_r}^{n} x_{i_1} .. x_{i_r}, \quad r = 1, 2, \cdots \tag{4.5}$$

and the *power sums*, or s-functions, as

$$s_r = (r) = \sum_{i=1}^{n} x_i^r, \quad r = 1, 2, \cdots \tag{4.6}$$

A special case of the a-function is the *augmented* unitary symmetric function, u_r (David et al., 1966):

$$u_r = [\ 1^r\ ] = r!(\ 1^r\ ) = r!a_r = \sum x_{i_1}..x_{i_r}\ , \tag{4.7}$$

summed over all ordered sets $i_1, \ldots, i_r$.

4.1.3 Recursive Expressions for Symmetric Functions

4.1.3.1 Interexpressibility Tables

Roe (1918) compiled comprehensive interexpressibility tables relating the various classes of symmetric functions to one another. These consist of a matrix of coefficients to be used in a polynomial which might yield, for example, $u_r = f(s_r)$. Of interest to this work is her relationship between the a-functions (often called *elementary* functions) and the s-functions; it is presented here in a form more clearly expressed by David et al. (1966):

$$a_r = \frac{1}{r!}u_r = \sum_{m=1}^{\lambda} \sum_{P} \frac{(-1)^{(r+m)}}{\pi_1!...\pi_\lambda!} \frac{s_{p_1}^{\pi_1}...s_{p_\lambda}^{\pi_\lambda}}{p_1^{\pi_1}...p_\lambda^{\pi_\lambda}} \tag{4.8}$$

David et al. (1966) also used this equation to construct interexpressibility tables describing polynomials in power-sum series (s) for a-functions up to and including weights of 12. For instance, a unitary symmetric function of weight 3 would be represented by the following polynomial from their Table 1.5.3:

$$a_3 = \frac{1}{3!}\left[\ (1)^3 - 3(2)(1) + 2(3)\ \right] \tag{4.9}$$

which, in terms of *s*-functions, is:

$$a_3 = \frac{1}{3!}\left[s_1^3 - 3s_2s_1 + 2s_3 \right] \tag{4.10}$$

and, when written as power sums, becomes:

$$a_3 = \frac{1}{3!}\left[\left[\sum_{i=1}^{n} x_i \right]^3 - 3\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} x_i + 2\sum_{i=1}^{n} x_i^3 \right] \tag{4.11}$$

However, neither Equation (4.8) nor Equation (4.11) is conducive to solution by computer without a tremendous table look-up effort.

4.1.3.2 Determinant Form

Fortunately, David et al. (1966) present another relationship between a_r and s_r in determinant form:

$$a_r = \frac{1}{r!}u_r = \frac{1}{r!}\det \begin{bmatrix} s_1 & 1 & 0 & 0 & \cdots & 0 \\ s_2 & s_1 & 2 & 0 & \cdots & 0 \\ s_3 & s_2 & s_1 & 3 & \cdots & 0 \\ s_4 & s_3 & s_2 & s_1 & \cdots & 0 \\ . & . & . & . & \cdots & r-1 \\ s_r & s_{r-1} & s_{r-2} & s_{r-3} & \cdots & s_1 \end{bmatrix} = \frac{\det \mathbf{M}}{r!} \tag{4.12}$$

This provides a practical method of calculating a_r that is also readily programmable on a computer.

4.2 Search for a Recursive Expression for the Vapor Fraction

Armed with a working knowledge of symmetric functions, we can manipulate the α equation developed in Chapter 3:

$$\sum_{r=0}^{N-1} \alpha^r \left\{ \sum_{i=1}^{N} z_i \; C_i \; a_r(C_1,\ldots,\hat{C}_i,\ldots,C_N) \right\} = 0 \tag{3.43}$$

to eliminate the exclusion term $\hat{C}_i$ and expand the symmetric functions into a more recognizable form. We will examine the results for several values of r and use them to write a general expression for α as a function of N.

4.2.1 Case I: r = N−2

Equation (3.43) yields the following coefficient for α:

$$\alpha^{(N-2)} \left\{ \sum_{i=1}^{N} z_i \; C_i \; a_{N-2}(C_1,\ldots,\hat{C}_i,\ldots,C_N) \right\} \tag{4.13}$$

We can expand the symmetric function a_{N-2} as shown in Equation (4.14). Since the exclusion of C_i from the product on the RHS gives $(N-1)$ terms, we must subtract $\frac{1}{C_i}$ from the sum to yield $(N-2)$:

$$a_{N-2}(C_1,\ldots,\hat{C}_i,\ldots,C_N) = (C_1 \cdots \hat{C}_i \cdots C_N)\left[\frac{1}{C_1} + \frac{1}{C_2} + \cdots + \frac{1}{C_N} - \frac{1}{C_i}\right] \tag{4.14}$$

To eliminate $\hat{C}_i$, we can write the product on the RHS of Equation (4.14) as

$$(C_1 \cdots \hat{C}_i \cdots C_N) = \frac{(C_1 \cdots C_N)}{C_i} \tag{4.15}$$

This maneuver will allow the summation to proceed over all N components.

After substituting Equation (4.15) into Equation (4.14), we have:

$$a_{N-2}(C_1,\dots,\hat{C}_i,\dots,C_N) = \frac{(C_1 \cdots C_N)}{C_i}\left[\frac{1}{C_1}+\frac{1}{C_2}+\cdots+\frac{1}{C_N}-\frac{1}{C_i}\right] \tag{4.16}$$

Substituting Equation (4.16) into Equation (4.13), cancelling C_i, multiplying by z_i, and then summing over i gives:

$$\alpha^{(N-2)}\left\{(C_1 \cdots C_N)\left[\left[\sum_{i=1}^{N} z_i\right]\left[\frac{1}{C_1}+\frac{1}{C_2}+\cdots+\frac{1}{C_N}\right]-\sum_{i=1}^{N}\frac{z_i}{C_i}\right]\right\} \tag{4.17}$$

We recall that $\sum_{i=1}^{N} z_i = 1$ and recognize that $(C_1 \cdots C_N) = \prod_{k=1}^{N} C_k$. Noting the presence of an elementary symmetric function in $\left[\frac{1}{C_1}+\cdots+\frac{1}{C_N}\right]$, we can write Equation (4.17) as:

$$\alpha^{(N-2)}\left\{\left[\prod_{k=1}^{N} C_k\right]\left[a_1\left[\frac{1}{C_1},\dots,\frac{1}{C_N}\right]-\sum_{i=1}^{N}\frac{z_i}{C_i}\right]\right\} \tag{4.18}$$

4.2.2 Case II: r = N−3

Equation (3.43) now becomes:

$$\alpha^{(N-3)}\left\{\sum_{i=1}^{N} z_i\, C_i\, a_{N-3}\,(C_1,\dots,\hat{C}_i,\dots,C_N)\right\} \tag{4.19}$$

We can expand the symmetric function a_{N-3} as shown in Equation (4.20). We eliminate $\hat{C}_i$ in the same manner as in Equation (4.15) and remove $\frac{1}{C_i}$ in a similar fashion. But this also deletes the term $\frac{1}{C_i^2}$ which is necessary to cancel the corresponding term in the product. Therefore, we must compensate by adding $\frac{1}{C_i^2}$:

$$a_{N-3}(C_1,\ldots,\hat{C}_i,\ldots,C_N) = \frac{(C_1\cdots C_N)}{C_i}\left[\frac{1}{C_1C_2}+\frac{1}{C_1C_3}+\cdots\right.$$

$$\left.+\frac{1}{C_{N-1}C_N}-\frac{1}{C_i}\left[\frac{1}{C_1}+\frac{1}{C_2}+\cdots+\frac{1}{C_N}\right]+\frac{1}{C_i^2}\right] \tag{4.20}$$

Substituting Equation (4.20) into Equation (4.19) and making consolidations similar to the previous development yields:

$$\alpha^{(N-3)}\left\{\left[\prod_{k=1}^{N}C_k\right]\left[a_2\left[\frac{1}{C_1},\ldots,\frac{1}{C_N}\right]-a_1\left[\frac{1}{C_1},\ldots,\frac{1}{C_N}\right]\sum_{i=1}^{N}\frac{z_i}{C_i}+\sum_{i=1}^{N}\frac{z_i}{C_i^2}\right]\right\} \tag{4.21}$$

4.2.3 Case III: r = 0

We have saved consideration of this case for last because the properties of a_0 are not readily apparent. It would seem reasonable to interpret $a_0(C_1,\ldots,\hat{C}_i,\ldots,C_N)$ as meaning "taking permutations of the elements of a set *zero* terms at a time." However, when $r = 0$, $\alpha^r \rightarrow 1$ and we know from previous developments that our α-polynomial does have a constant term. Therefore, $a_0(C_1,\ldots,\hat{C}_i,\ldots,C_N)$ must equal *one*, after Macdonald (1979). So, for $r = 0$, Equation (3.44) becomes:

$$\left\{\sum_{i=1}^{N} z_i C_i\right\} \tag{4.24}$$

We show Equation (4.18) and Equation (4.21) again to look for patterns that may assist us in writing the expression for the $(N-p)$-th term:

$$\alpha^{(N-2)}\left\{\left[\prod_{k=1}^{N} C_k\right]\left[a_1\left[\frac{1}{C_1},\ldots,\frac{1}{C_N}\right]-\sum_{i=1}^{N}\frac{z_i}{C_i}\right]\right\} \tag{4.18}$$

$$\alpha^{(N-3)}\left\{\left[\prod_{k=1}^{N} C_k\right]\left[a_2\left[\frac{1}{C_1},\ldots,\frac{1}{C_N}\right]-a_1\left[\frac{1}{C_1},\ldots,\frac{1}{C_N}\right]\sum_{i=1}^{N}\frac{z_i}{C_i}+\sum_{i=1}^{N}\frac{z_i}{C_i^2}\right]\right\} \tag{4.21}$$

4.2.4 The General Case: r = (N−p), p = 1, 2, ..., N

After substituting for r, Equation (3.43) becomes:

$$\sum_{p=1}^{N}\alpha^{(N-p)}\left\{\sum_{i=1}^{N} z_i\ C_i\ a_{N-p}(C_1,\ldots,\hat{C}_i,\ldots,C_N)\right\} = 0 \tag{4.23}$$

By continuing the expansion of this equation in the same fashion as in the first two cases, we note a descending order of the symmetric function and an ascending exponent of C_i with each additional term. This leads to a general expression:

$$\sum_{p=1}^{N}\alpha^{(N-p)}\left\{\sum_{i=1}^{N} z_i\ (C_1\cdots C_N)\Big[a_{p-1}(C_1^{-1},\ldots,C_N^{-1})\ -\right.$$

$$C_i^{-1}a_{p-2}(C_1^{-1},\ldots,C_N^{-1})+C_i^{-2}a_{p-3}(C_1^{-1},\ldots,C_N^{-1})-C_i^{-3}a_{p-4}(C_1^{-1},\ldots,C_N^{-1})$$

$$\left.+\ \cdots\ \pm C_i^{-(p-2)}a_1(C_1^{-1},\ldots,C_N^{-1})\ \pm C_i^{-(p-1)}\Big]\right\} = 0 \tag{4.24}$$

Multiply by z_i, sum i from 1 to N and recall that $(C_1 \cdots C_N) = \prod_{k=1}^{N} C_k$:

$$\sum_{p=1}^{N} \alpha^{(N-p)} \left\{ \left[\prod_{k=1}^{N} C_k \right] \left[a_{p-1} - a_{p-2} \sum_{i=1}^{N} \frac{z_i}{C_i} + a_{p-3} \sum_{i=1}^{N} \frac{z_i}{C_i^2} - \cdots + \right. \right.$$

$$\left. \left. (-1)^{(p-1)} \sum_{i=1}^{N} \frac{z_i}{C_i^{(p-1)}} \right] \right\} = 0 \tag{4.25}$$

Since $\prod_{k=1}^{N} C_k$ does not involve p, we can move this term outside the summation sign and then divide it out as a factor common to all powers of α. By examining the relationship between p, the subscripts of a and the superscripts of C_i, we can collapse Equation (4.25) into a more compact form:

$$\sum_{p=1}^{N} \alpha^{(N-p)} \left\{ \sum_{j=1}^{p} \left[(-1)^{j+1} (a_{p-j}) \sum_{i=1}^{N} \frac{z_i}{C_i^{j-1}} \right] \right\} = 0 \tag{4.26}$$

$$\text{where } a_{p-j} = a_{p-j}(C_1^{-1}, \ldots, C_N^{-1}) \tag{4.27}$$

$$a_0 = 1 \tag{4.28}$$

$$C_i = (K_i)_{T,P} - 1 \tag{4.29}$$

4.3 A Generalized Multicomponent Equation for the Vapor Fraction

Using the determinant expression for the elementary symmetric functions that was presented in **4.1.3.2**, Equation (4.27) becomes:

$$a_{p-j} = a_{p-j}(\, C_1^{-1}, \ldots, C_N^{-1}\,) = \frac{\det \mathbf{M}}{(p-j)!} \tag{4.30}$$

The matrix **M** has dimensions $(p-j) \times (p-j)$ and elements given by:

$$[m_{kl}] = \begin{cases} s_{k-l+1} & \text{if } l \leq k \\ k & \text{if } l = k+1 \\ 0 & \text{if } l > k+1 \end{cases} \tag{4.31}$$

The s elements are given by $s_\lambda = \sum_{i=1}^{N} \left[\frac{1}{C_i} \right]^\lambda, \; \lambda = 1, 2, \ldots, (p-j)$ (4.32)

Chapter 5

VALIDATION OF THE GENERALIZED EQUATION

The first test of validity for Equation (4.26) requires that it be equivalent to the form of the objective function presented in Equation (3.40). Second, it must generate the same coefficients for the α polynomial that were produced through the expansion of the objective function in Equation (3.25) through Equation (3.32). Third, the equation must predict the proper vapor fraction for a fluid undergoing an isothermal, isobaric flash process.

The first test is supplied by a mathematical proof in Appendix A. The second test can be accomplished by comparing the coefficients produced by Equation (4.26) with those of Equation (3.33). Since this equation has already been shown to reduce to that for a quaternary system under the proper constraints on z_5 and K_5, then, by induction, we can state that the polynomial produced by Equation (4.26) will do the same and therefore should be valid for any number of components.

The third test will be satisfied by comparing the equilibrium ratios generated by Equation (4.26) with experimental values determined for several multicomponent hydrocarbon fluids.

5.1 The Generalized α Equation for a Quinary System

For a five-component system, Equation (4.26) becomes:

$$\sum_{p=1}^{5} \alpha^{(5-p)} \left\{ \sum_{j=1}^{p} \left[(-1)^{j+1} a_{p-j} \sum_{i=1}^{5} \frac{z_i}{C_i^{\,j-1}} \right] \right\} = 0 \qquad (5.1)$$

which will yield a quartic polynomial in α:

$$\mu_4\alpha^4 + \mu_3\alpha^3 + \mu_2\alpha^2 + \mu_1\alpha + \mu_0 = 0 \tag{5.2}$$

5.1.1 Coefficient μ_4 (p = 1)

$$\mu_4 = a_0(C_i^{-1}) \; [\, z_1 + z_2 + z_3 + z_4 + z_5 \,] \tag{5.3}$$

We have already said that $a_0(C_i^{-1})$ is defined as one and the sum of the mole fractions also equals one, so Equation (5.3) yields:

$$\mu_4 = 1 \tag{5.4}$$

5.1.2 Coefficient μ_3 (p = 2)

$$\mu_3 = a_1(C_i^{-1}) \sum_{i=1}^{5} z_i - a_0(C_i^{-1}) \sum_{i=1}^{5} \frac{z_i}{C_i} \tag{5.5}$$

$$\mu_3 = \left[\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \frac{1}{C_4} + \frac{1}{C_5} \right](1) -$$

$$(1)\left[\frac{z_1}{C_1} + \frac{z_2}{C_2} + \frac{z_3}{C_3} + \frac{z_4}{C_4} + \frac{z_5}{C_5} \right] \tag{5.6}$$

$$\mu_3 = \frac{1 - z_1}{C_1} + \frac{1 - z_2}{C_2} + \frac{1 - z_3}{C_3} + \frac{1 - z_4}{C_4} + \frac{1 - z_5}{C_5} \tag{5.7}$$

5.1.3 Coefficient μ_2 (p = 3)

$$\mu_2 = a_2(C_i^{-1}) \sum_{i=1}^{5} z_i - a_1(C_i^{-1}) \sum_{i=1}^{5} \frac{z_i}{C_i} + a_0(C_i^{-1}) \sum_{i=1}^{5} \frac{z_i}{C_i^2} \tag{5.8}$$

$$\begin{aligned}\mu_2 = &\left[\frac{1}{C_1C_2} + \frac{1}{C_1C_3} + \frac{1}{C_1C_4} + \frac{1}{C_1C_5} + \frac{1}{C_2C_3} + \frac{1}{C_2C_4} + \frac{1}{C_2C_5} + \frac{1}{C_3C_4} \right.\\ &\left. + \frac{1}{C_3C_5} + \frac{1}{C_4C_5} \right](1) - \left[\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \frac{1}{C_4} + \frac{1}{C_5} \right] \left[\frac{z_1}{C_1} + \frac{z_2}{C_2} + \frac{z_3}{C_3} \right.\\ &\left. + \frac{z_4}{C_4} + \frac{z_5}{C_5} \right] + (1)\left[\frac{z_1}{C_1^2} + \frac{z_2}{C_2^2} + \frac{z_3}{C_3^2} + \frac{z_4}{C_4^2} + \frac{z_5}{C_5^2} \right]\end{aligned} \tag{5.9}$$

$$\begin{aligned}\mu_2 = &\frac{1}{C_1C_2} + \frac{1}{C_1C_3} + \frac{1}{C_1C_4} + \frac{1}{C_1C_5} + \frac{1}{C_2C_3} + \frac{1}{C_2C_4} + \frac{1}{C_2C_5} + \frac{1}{C_3C_4} + \\ &\frac{1}{C_3C_5} + \frac{1}{C_4C_5} - \frac{z_1}{C_1^2} - \frac{z_2}{C_2^2} - \frac{z_3}{C_3^2} - \frac{z_4}{C_4^2} - \frac{z_5}{C_5^2} - \frac{z_1 + z_2}{C_1C_2} - \frac{z_1 + z_3}{C_1C_3} - \\ &\frac{z_1 + z_4}{C_1C_4} - \frac{z_1 + z_5}{C_1C_5} - \frac{z_2 + z_3}{C_2C_3} - \frac{z_2 + z_4}{C_2C_4} - \frac{z_2 + z_5}{C_2C_5} - \frac{z_3 + z_4}{C_3C_4} - \frac{z_3 + z_5}{C_3C_5} - \\ &\frac{z_4 + z_5}{C_4C_5} + \frac{z_1}{C_1^2} + \frac{z_2}{C_2^2} + \frac{z_3}{C_3^2} + \frac{z_4}{C_4^2} + \frac{z_5}{C_5^2}\end{aligned} \tag{5.10}$$

$$\begin{aligned}\mu_2 = &\frac{1 - z_1 - z_2}{C_1C_2} + \frac{1 - z_1 - z_3}{C_1C_3} + \frac{1 - z_1 - z_4}{C_1C_4} + \frac{1 - z_1 - z_5}{C_1C_5} + \frac{1 - z_2 - z_3}{C_2C_3} + \\ &\frac{1 - z_2 - z_4}{C_2C_4} + \frac{1 - z_2 - z_5}{C_2C_5} + \frac{1 - z_3 - z_4}{C_3C_4} + \frac{1 - z_3 - z_5}{C_3C_5} + \frac{1 - z_4 - z_5}{C_4C_5}\end{aligned} \tag{5.11}$$

5.1.4 Coefficient μ_1 (p = 4)

$$\mu_1 = a_3(C_i^{-1})\sum_{i=1}^{5} z_i - a_2(C_i^{-1})\sum_{i=1}^{5} \frac{z_i}{C_i} + a_1(C_i^{-1})\sum_{i=1}^{5} \frac{z_i}{C_i^2} - a_0(C_i^{-1})\sum_{i=1}^{5} \frac{z_i}{C_i^3} \qquad (5.12)$$

$$\begin{aligned}
\mu_1 = {} & \left[\frac{1}{C_1C_2C_3} + \frac{1}{C_1C_2C_4} + \frac{1}{C_1C_2C_5} + \frac{1}{C_1C_3C_4} + \frac{1}{C_1C_3C_5} + \right. \\
& \left. \frac{1}{C_1C_4C_5} + \frac{1}{C_2C_3C_4} + \frac{1}{C_2C_3C_5} + \frac{1}{C_2C_4C_5} + \frac{1}{C_3C_4C_5} \right] (1) - \\
& \left[\frac{1}{C_1C_2} + \frac{1}{C_1C_3} + \frac{1}{C_1C_4} + \frac{1}{C_1C_5} + \frac{1}{C_2C_3} + \frac{1}{C_2C_4} + \frac{1}{C_2C_5} + \frac{1}{C_3C_4} + \right. \\
& \left. \frac{1}{C_3C_5} + \frac{1}{C_4C_5} \right] \left[\frac{z_1}{C_1} + \frac{z_2}{C_2} + \frac{z_3}{C_3} + \frac{z_4}{C_4} + \frac{z_5}{C_5} \right] + \\
& \left[\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3} + \frac{1}{C_4} + \frac{1}{C_5} \right] \left[\frac{z_1}{C_1^2} + \frac{z_2}{C_2^2} + \frac{z_3}{C_3^2} + \frac{z_4}{C_4^2} + \frac{z_5}{C_5^2} \right] - \\
& (1) \left[\frac{z_1}{C_1^3} + \frac{z_2}{C_2^3} + \frac{z_3}{C_3^3} + \frac{z_4}{C_4^3} + \frac{z_5}{C_5^3} \right] \qquad (5.13)
\end{aligned}$$

It is evident that the $\dfrac{z_i}{C_i^2 C_j}$ terms in the second part of Equation (5.13) will cancel those in the third part, while the $\dfrac{z_i}{C_i^3}$ terms in the third part will negate the entire fourth part of the equation. The first and second parts yield:

$$\mu_1 = \frac{1 - z_1 - z_2 - z_3}{C_1C_2C_3} + \frac{1 - z_1 - z_2 - z_4}{C_1C_2C_4} + \frac{1 - z_1 - z_2 - z_5}{C_1C_2C_5} + \frac{1 - z_1 - z_3 - z_4}{C_1C_3C_4}$$

$$+ \frac{1 - z_1 - z_3 - z_5}{C_1 C_3 C_5} + \frac{1 - z_1 - z_4 - z_5}{C_1 C_4 C_5} + \frac{1 - z_2 - z_3 - z_4}{C_2 C_3 C_4} + \frac{1 - z_2 - z_3 - z_5}{C_2 C_3 C_5} +$$

$$\frac{1 - z_2 - z_4 - z_5}{C_2 C_4 C_5} + \frac{1 - z_3 - z_4 - z_5}{C_3 C_4 C_5} \tag{5.14}$$

5.1.5 Coefficient μ_0 (p = 5)

$$\mu_0 = a_4(C_i^{-1}) \sum_{i=1}^{5} z_i - a_3(C_i^{-1}) \sum_{i=1}^{5} \frac{z_i}{C_i} + a_2(C_i^{-1}) \sum_{i=1}^{5} \frac{z_i}{C_i^2} -$$

$$a_1(C_i^{-1}) \sum_{i=1}^{5} \frac{z_i}{C_i^3} + a_0(C_i^{-1}) \sum_{i=1}^{5} \frac{z_i}{C_i^4} \tag{5.15}$$

The analogous cancellations of the higher-order $\frac{1}{C_i}$ terms will occur, leaving a sum of five terms having the form

$$\frac{1 - z_i - z_j - z_k - z_l}{C_i \; C_j \; C_k \; C_l} \tag{5.16}$$

Since the mole fractions must sum to one, we can replace the numerator of Equation (5.16) with the mole fraction of the remaining component, z_m , to yield:

$$\mu_0 = \frac{z_1}{C_2 C_3 C_4 C_5} + \frac{z_2}{C_1 C_3 C_4 C_5} + \frac{z_3}{C_1 C_2 C_4 C_5} +$$

$$\frac{z_4}{C_1 C_2 C_3 C_5} + \frac{z_5}{C_1 C_2 C_3 C_4} \tag{5.17}$$

A term-by-term comparison with Equation (3.32) shows that the generalized α polynomial [Equation (4.26)] produces identical results.

5.2 Reproduction of Experimental Vapor-Liquid Equilibrium Data

5.2.1 Flash Calculation Package

The flash calculation package used in this work incorporates the K-value equation of Wilson (1969) and the modified PREOS proposed by Stryjek and Vera (1986a). An attempt was made to use the K-value prediction of Varotsis (1989) but, as noted in Chapter 2, it was developed to characterize a broad-spectrum petroleum reservoir condensate or crude oil. It experiences difficulty handling an arbitrary hydrocarbon mixture, such as the artificial systems for which equilibrium data is available to validate this work.

The volume correction of Mathias et al. (1989) and the complementary calculation of Schick and Prausnitz (1968) for mixture pseudo-critical volume are incorporated into the PRSV EOS but since it is only required to generate compressibility factors and fugacities, the modifications have no noticeable effect on the computations. The PRSV EOS shows marked improvement over the PREOS when used to duplicate bubblepoint and dewpoint studies performed by Warren (1991).

The binary interaction coefficients used in the PRSV EOS are taken from Knapp et al. (1982) and Walas (1985). Physical property data and equation parameters for the chemical components are extracted from Stryjek and Vera (1986b,c), Kumar (1987) and Proust and Vera (1989).

The computation of the determinant used to generate the elementary symmetric functions is accomplished with a modified Gaussian elimination routine. The first elementary symmetric function, a_1, is defined by a $[1\times1]$ matrix, whose determinant constitutes the element itself. By definition, a_0 is set equal to one.

The polynomial is evaluated at the bubblepoint line ($\alpha = 0$) and an interval-halving technique is used to march across the two-phase region until the value of the polynomial changes sign, indicating the vicinity of the root. Then a Newton-Raphson iterative search is conducted to converge to the exact value of α.

5.2.2 Binary System

The fugacity-based flash algorithm is used to replicate the equilibrium ratios determined by Bloomer et al. (1953) for a methane-ethane system at a temperature of −60 °F over a pressure range of 100-900 psia. Comparisons of calculated and empirical values of K_{CH_4} and $K_{C_2H_6}$ appear in Figure 5.1 and 5.2, respectively. The results lie within the margin of error attributable to the PRSV EOS.

5.2.3 Septenary System

Standing (1977) provides a sample flash calculation for a seven-component hydrocarbon system reported by Dodson and Standing (1941), complete with values for experimental K_i and the vapor fraction. This sort of data allows the calculation of α solely on the basis of computing the coefficients of the α-polynomial and determining the applicable root, with no recourse to the equation of state. Once the interval-halving search provides an initial estimate of the root, the Newton-Raphson technique converges in three iterations to a value of α identical to that calculated by Standing.

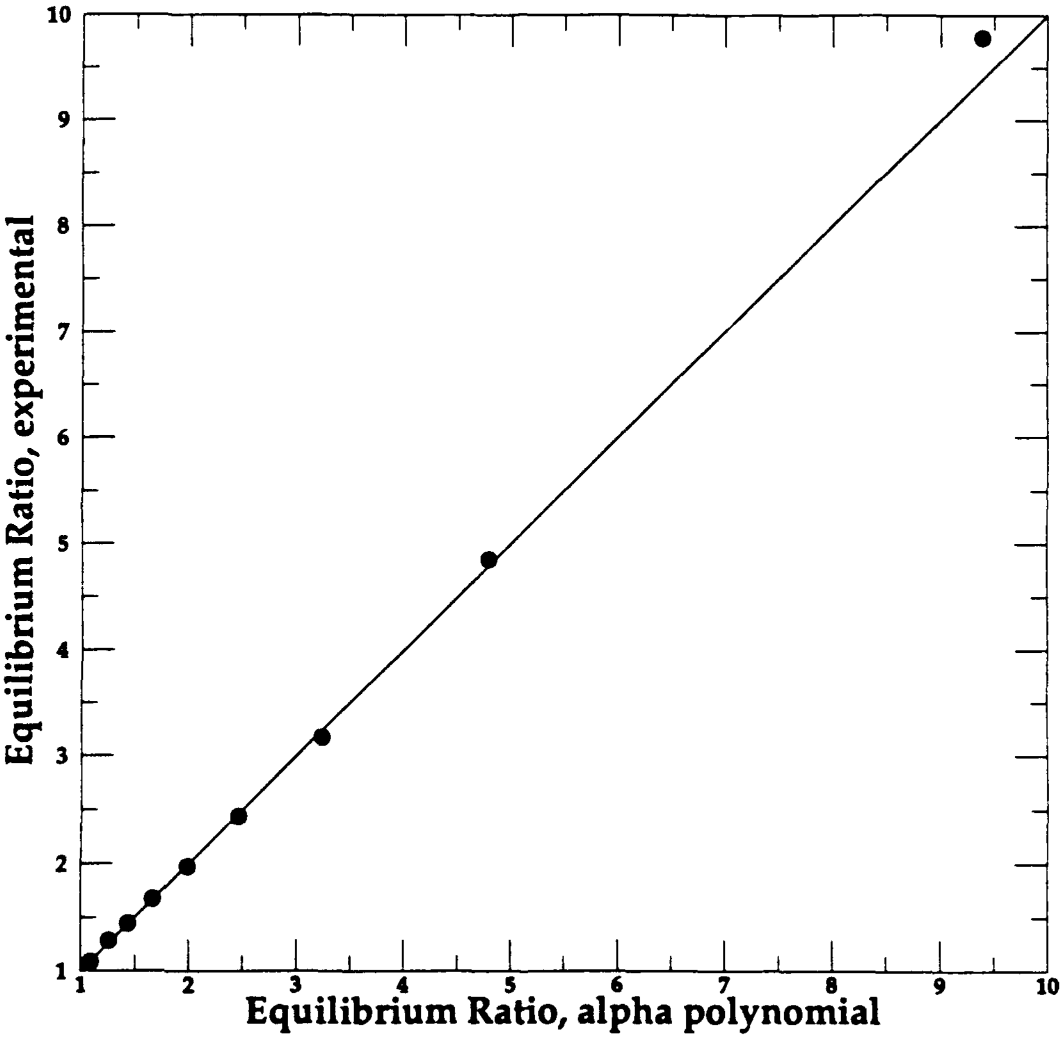


Figure 5.1: A Comparison of Predicted and Experimental Equilibrium Ratios for Methane

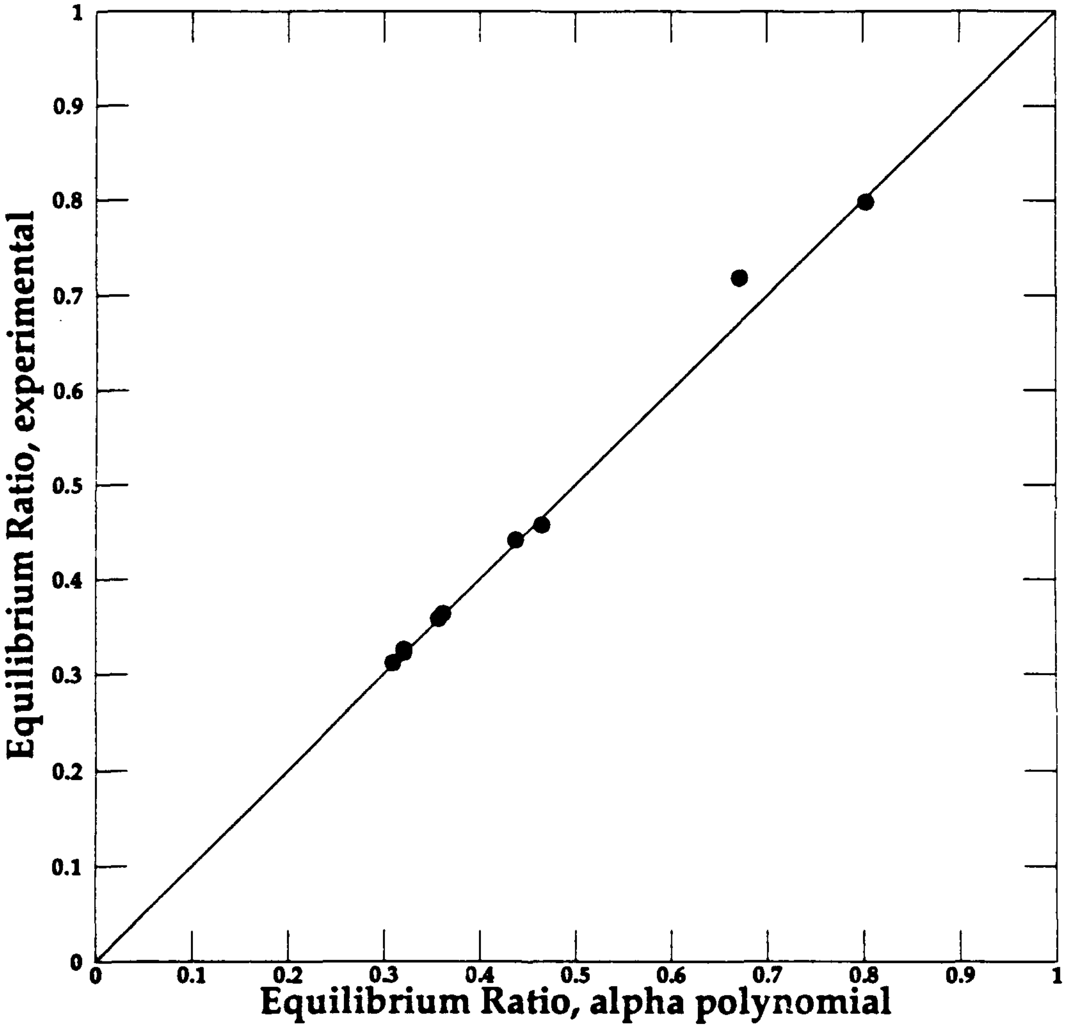


Figure 5.2: A Comparison of Predicted and Experimental Equilibrium Ratios for Ethane

5.2.4 Predicting Roots with the Fourier-Budan Theorem

A useful theorem for predicting the number of roots of a polynomial that can occur on a particular interval is that of Fourier and Budan (Barbeau, 1989). Suppose $p(t)$ is a polynomial over the field of real numbers, **R**, and that u and v are real numbers with $u < v$ and $p(u) \cdot p(v) \neq 0$. The number of zeros between u and v cannot be greater than $A - B$, where A is the number of changes of sign in the sequence $\{ p(u), p'(u), p''(u), \ldots, p^{(n)}(u) \}$ and B is the number of changes of sign in the sequence $\{ p(v), p'(v), p''(v), \ldots, p^{(n)}(v) \}$. If this number differs from $A - B$, it must do so by an even amount.

An interesting aspect of the polynomial expression for the vapor fraction is its capability to mathematically confirm the existence of a unique value within the two-phase region for a particular set of feed conditions. This is equivalent to stating that the polynomial has only one zero on the interval $0 \leq \alpha \leq 1$. From the physics of the problem, we know this to be true but, by the use of the Fourier-Budan theorem, we can also prove it rigorously.

Let us test this theorem on the septenary system of Standing (1977) utilized in **5.2.3**; this is represented by a sixth-order polynomial:

$$\mu_6\alpha^6 + \mu_5\alpha^5 + \mu_4\alpha^4 + \mu_3\alpha^3 + \mu_2\alpha^2 + \mu_1\alpha + \mu_0 = 0 \qquad (5.18)$$

where

$$\begin{array}{lrlr} \mu_0 = & -9.58519 & \mu_3 = & 87.24949 \\ \mu_1 = & 65.90501 & \mu_4 = & -21.71701 \\ \mu_2 = & -120.72959 & \mu_5 = & -1.76522 \end{array}$$

$$\mu_6 = 1.00000$$

We can differentiate Equation (5.18) six times and form the derivative sequences for $u = 0$ and $v = 1$. The sign changes are summarized in Table 5.1.

Table 5.1 - Derivative Series of Fourier-Budan Theorem:
7-Component Hydrocarbon System (Standing, 1977)

	$u = 0$	$v = 1$
$f(\alpha)$	−	+
$f'(\alpha)$	+	−
$f''(\alpha)$	−	+
$f'''(\alpha)$	+	+
$f^{(4)}(\alpha)$	−	−
$f^{(5)}(\alpha)$	−	−
$f^{(6)}(\alpha)$	+	+
	$A = 5$	$B = 4$

Since $A - B = 1$, there exists only a single root of the polynomial on the interval [0,1]; an examination of the graph of the function (Figure 5.3) confirms this fact. Therefore, we can use the interval-halving and Newton-Raphson solution procedures outlined at the beginning of this chapter with confidence that they will obtain a unique, realistic value of the vapor fraction.

5.2.5 Decenary System

Gregory et al. (1971) performed equilibrium measurements on a lean natural gas at cryogenic conditions. It is reported as a ten-component system with the feed composition shown in Table 5.2. This is a very "sparse" ten-component gas, with six components present in extremely dilute concentrations. The K-values for the last four constituents were zero for eleven of the sixteen operating conditions tested in this work, denoted in Table 5.3 by the run number assigned by the investigators. The remaining twelve sets of published data duplicate conditions in one of the tested runs or are incomplete due to apparatus failure. The use of the Fourier-Budan theorem provides warning that perhaps this gas would be better represented by an equivalent "lumped" system.

Recall that the number of roots predicted by the Fourier-Budan theorem is the maximum possible and may differ from the true value by only an even integer. This is demonstrated in Table 5.4, where both the predicted and actual number of roots for each run are tabulated. The Newton-Raphson technique converges to the experimental value for ten of the sixteen runs. Three other data points follow the proper trend, while no root is found on the interval [0,1] for three other conditions (Figure 5.4).

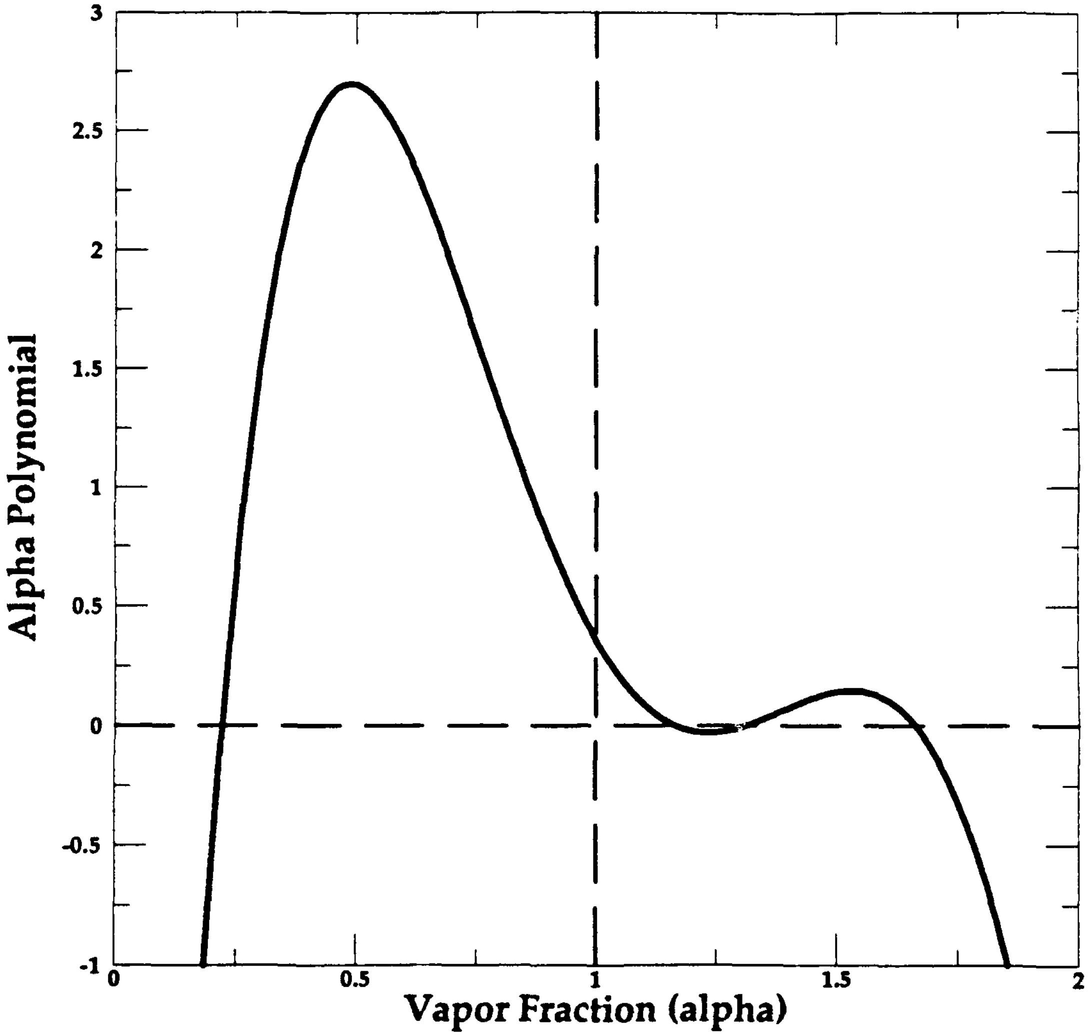


Figure 5.3: The Polynomial for a 7-Component Hydrocarbon System

**Table 5.2 - Feed Composition:
10-Component Natural Gas
(Gregory et al., 1971)**

Component	z_i	Component	z_i
Nitrogen	0.00600	*n*-Butane	0.00070
Methane	0.95970	*i*-Pentane	0.00030
Ethane	0.03000	*n*-Pentane	0.00010
Propane	0.00390	3-Methylpentane	0.00025
i-Butane	0.00070	2-Methylhexane	0.00015

Table 5.3 - Experimental Flash Conditions: 10-Component Natural Gas (Gregory et al., 1971)

Run	Pressure (psia)	Temperature (°F)	Run	Pressure (psia)	Temperature (°F)
1	300.0	-156.3	14	100.0	-200.0
3	100.0	-206.0	15	500.6	-127.0
4	700.0	-103.0	18	23.0	-252.0
7	500.0	-125.0	20	497.0	-129.0
8	498.5	-120.0	21	23.5	-251.5
9	695.0	-105.0	25	700.0	-107.0
10	100.0	-203.3	26	298.0	-157.5
12	100.0	-195.0	28	500.0	-130.0

Table 5.4 - Results of the α-Polynomial and Fourier-Budan Theorem:
10-Component Natural Gas
(Gregory et al., 1971)

Run	Root Limit (Actual)	Newton-Raphson Iterations	Roots on [0,1]		
			Initial Guess	Calculated	Experimental
1	2	3 3	0.605 0.825	0.603 0.822	0.603 0.603
3	2 (0)	*	*****	*****	0.155
4	2 (0)	*	*****	*****	0.911
7	3 (1)	4	0.775	0.772	0.761
8	1	3	0.905	0.904	0.908
9	2 (0)	*	*****	*****	0.795
10	4 (2)	4 4	0.695 0.915	0.692 0.912	0.687 0.687
12	2	4 5	0.895 0.965	0.891 0.966	0.890 0.890
14	3 (1)	4	0.835	0.830	0.830
15	2	3 3	0.735 0.845	0.737 0.843	0.747 0.747
18	4 (2)	3 3	0.045 0.415	0.044 0.415	0.109 0.109
20	4 (2)	3 3	0.585 0.775	0.587 0.773	0.591 0.591
21	2	4 3	0.015 0.645	0.011 0.642	0.078 0.078
25	1	3	0.385	0.380	0.548
26	5 (1)	3	0.435	0.434	0.430
28	3 (1)	4	0.485	0.480	0.486

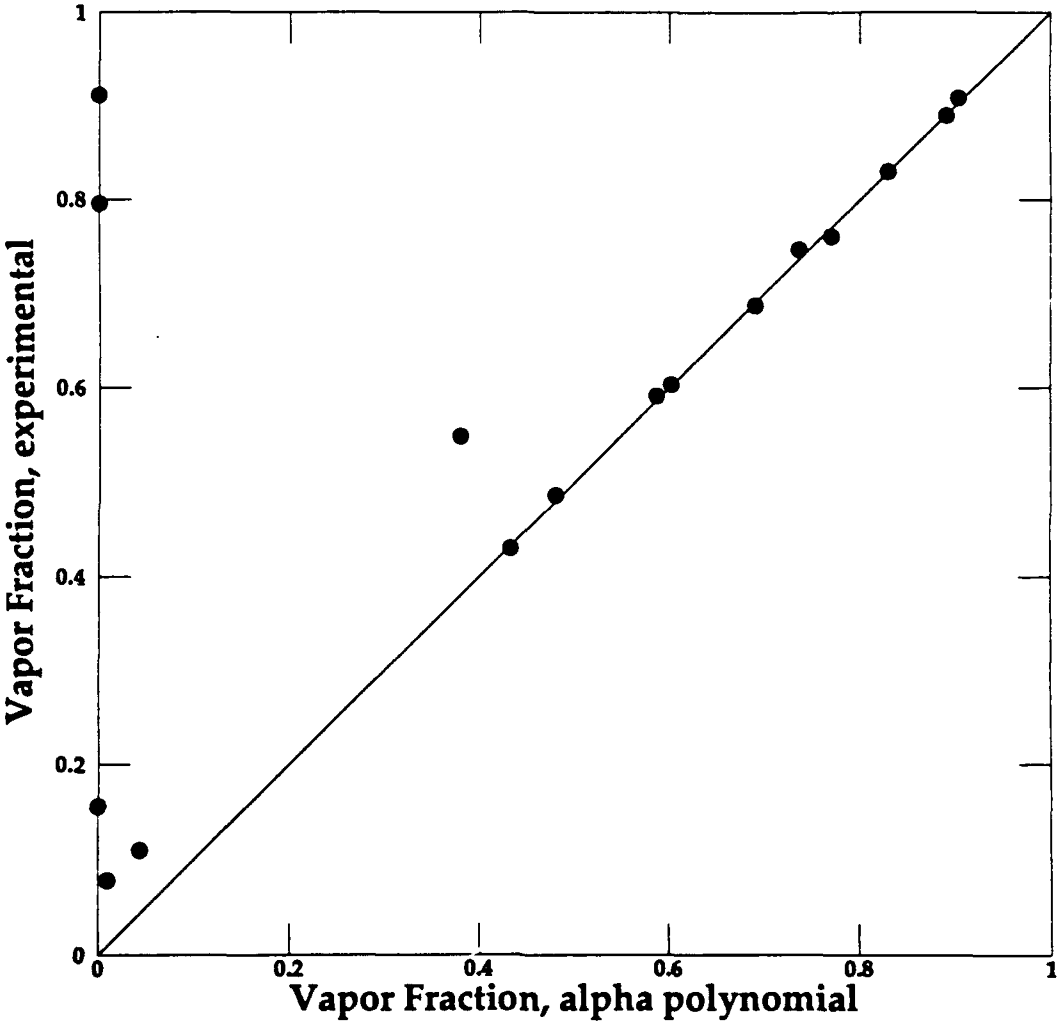


Figure 5.4: A Comparison of Predicted and Experimental Vapor Fractions for a 10-Component Natural Gas

An examination of graphs of the polynomial's behavior over a range of α for Runs 3 and 4 (Figures 5.5 and 5.6) confirms the algorithm's prediction that no roots exist within the phase envelope. The case of Run 9 is not so obvious. Its graph (Figure 5.7) shows that the function exists entirely above the abscissa; hence no root is possible. However, if the resolution of the graph is increased to examine the region very near the axis, two local minima are revealed (Figure 5.8). One of these corresponds to the experimental value of α determined for this run. The polynomial is attempting to represent the system's behavior but is not completely successful because the low concentration of certain components effectively prevents the gas from acting like a decenary system.

It is instructive to compare the form of the α-polynomial with that of the Rachford-Rice objective function which is superimposed on Figure 5.7. The same high-resolution scan of the graph of the latter equation depicts no equivalent maxima which might identify the vapor fraction in the manner of the polynomial.

5.2.6 Lumping a Decenary System into a Quaternary System

The α-polynomial successfully converges to the proper answer for a majority of the runs; however, it also yields multiple roots where the physics of the problem allows only one. This suggests that the system is not being properly modeled. The categorization of the fluid as a ten-component natural gas is overly generous in light of the fact that six of its chemical constituents are present in mole fractions measured in the ten-thousandths. It was decided to represent this sparse gas as a four-component lumped system, consisting of methane, ethane, nitrogen and propane.

The mole fractions of this new fluid are normalized and the resulting cubic polynomial in α is solved. The Fourier-Budan theorem predicts a maximum of one

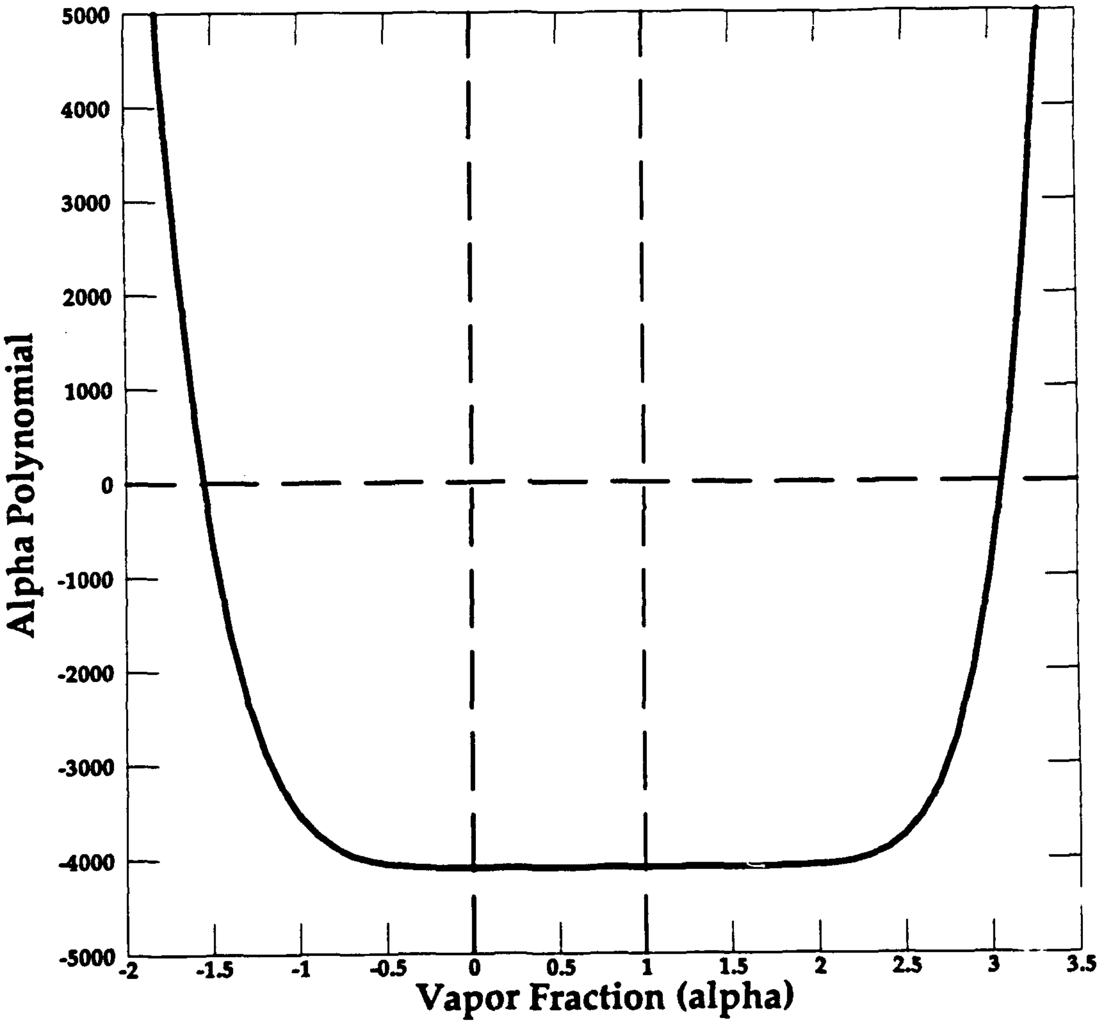


Figure 5.5: The Polynomial for a 10-Component Natural Gas [Run 3]

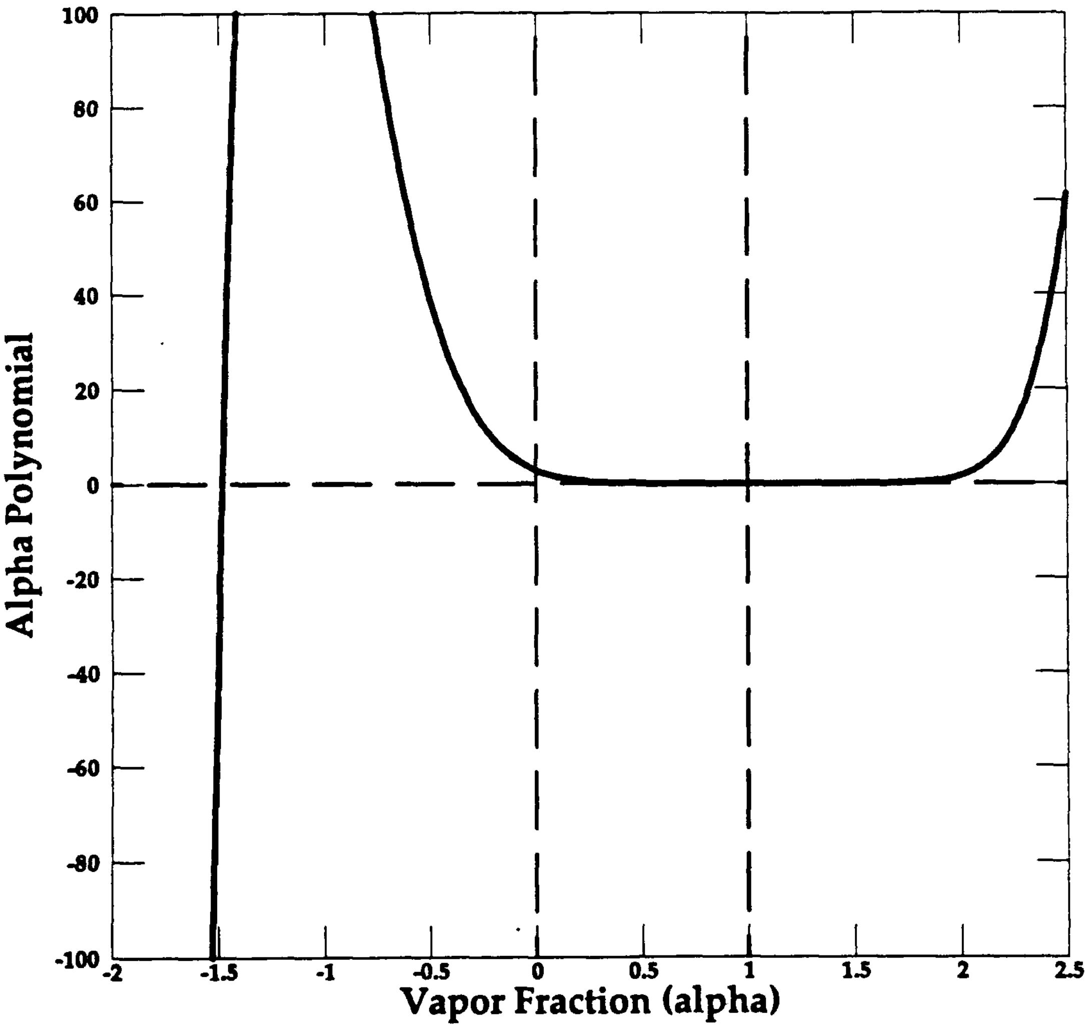


Figure 5.6: The Polynomial for a 10-Component Natural Gas [Run 4]

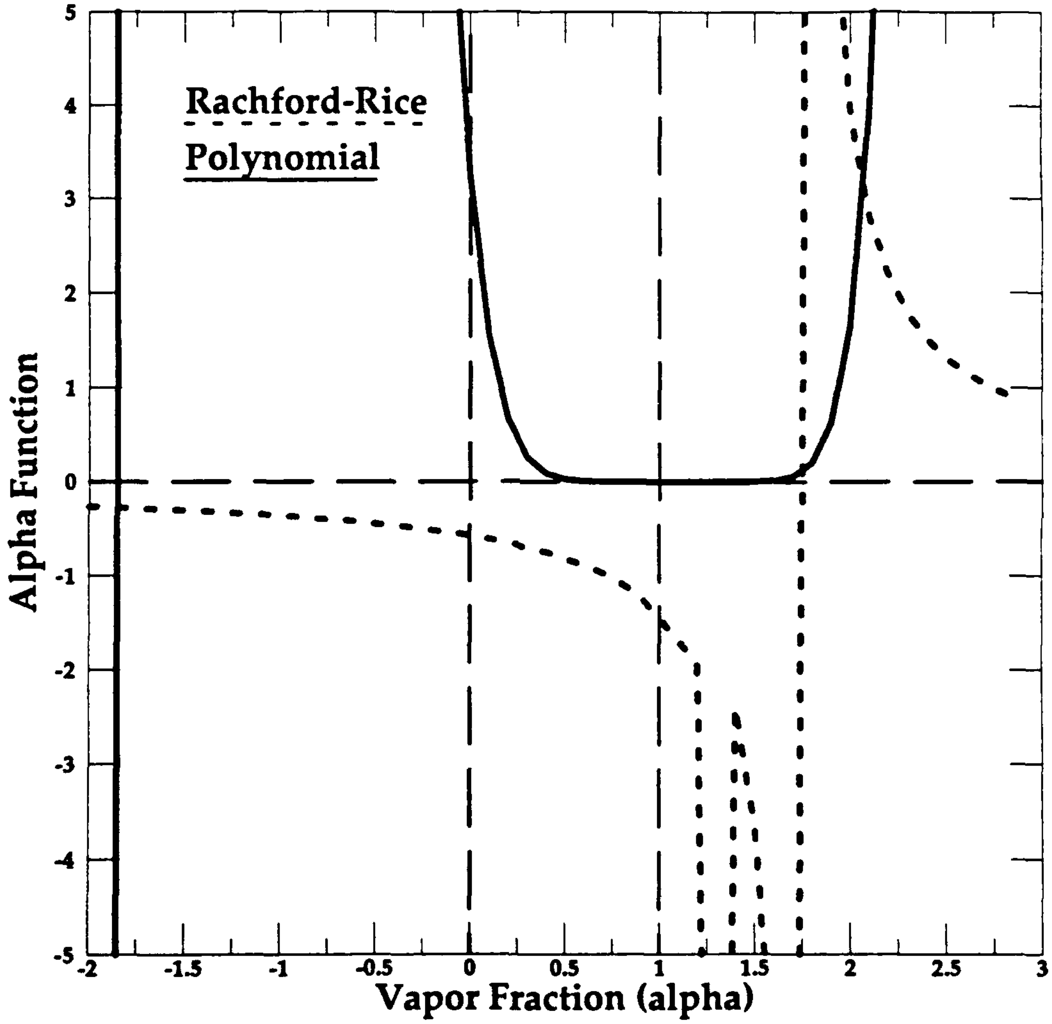


Figure 5.7: The Polynomial for a 10-Component Natural Gas [Run 9]

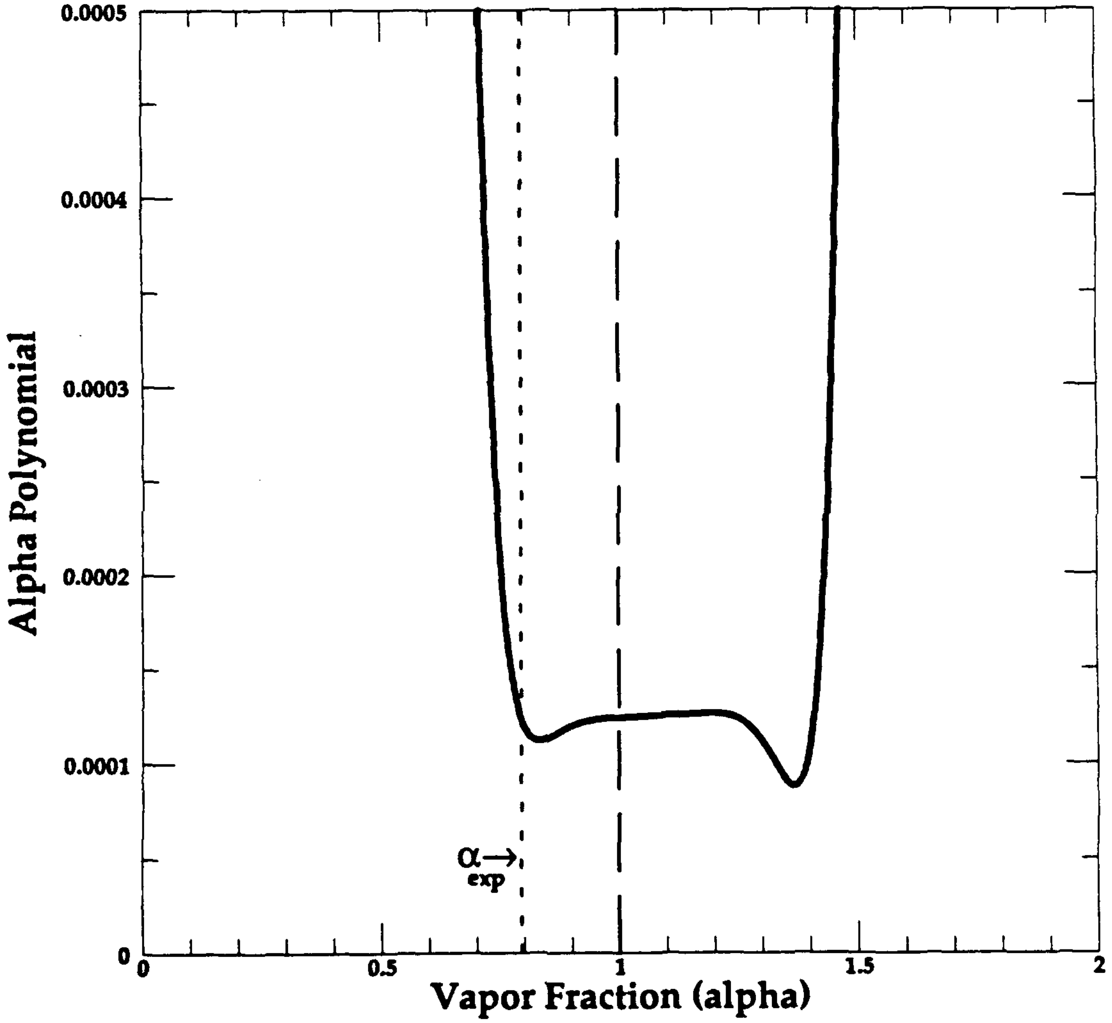


Figure 5.8: A High-Resolution View of the Polynomial for a 10-Component Natural Gas [Run 9]

root in the two-phase region, to which all sixteen runs converge. The numbers tabulated in Table 5.5 and displayed graphically in Figure 5.9 attest to the validity of this lumping scheme. An attempt was made to eliminate the next leanest component--propane--from the mixture and model the system as a ternary, but this resulted in spurious roots for all data runs and was hence rejected as unrealistic.

Table 5.5 - Results of the α-Polynomial and Fourier-Budan Theorem:
"Lumped" 4-Component Natural Gas
(Gregory et al., 1971)

Run	Root Limit (Actual)	Newton-Raphson Iterations	Roots on [0,1]		
			Initial Guess	Calculated	Experimental
1	1	3	0.635	0.631	0.603
3	1	3	0.035	0.038	0.155
4	1	3	0.995	0.998	0.911
7	1	3	0.815	0.819	0.761
8	1	3	0.935	0.932	0.908
9	1	3	0.925	0.920	0.795
10	1	3	0.715	0.711	0.687
12	1	3	0.895	0.899	0.890
14	1	3	0.845	0.842	0.830
15	1	3	0.765	0.770	0.747
18	1	3	0.035	0.037	0.109
20	1	3	0.635	0.631	0.591
21	1	3	0.015	0.012	0.078
25	1	3	0.775	0.779	0.548
26	1	3	0.475	0.473	0.430
28	1	3	0.535	0.536	0.486

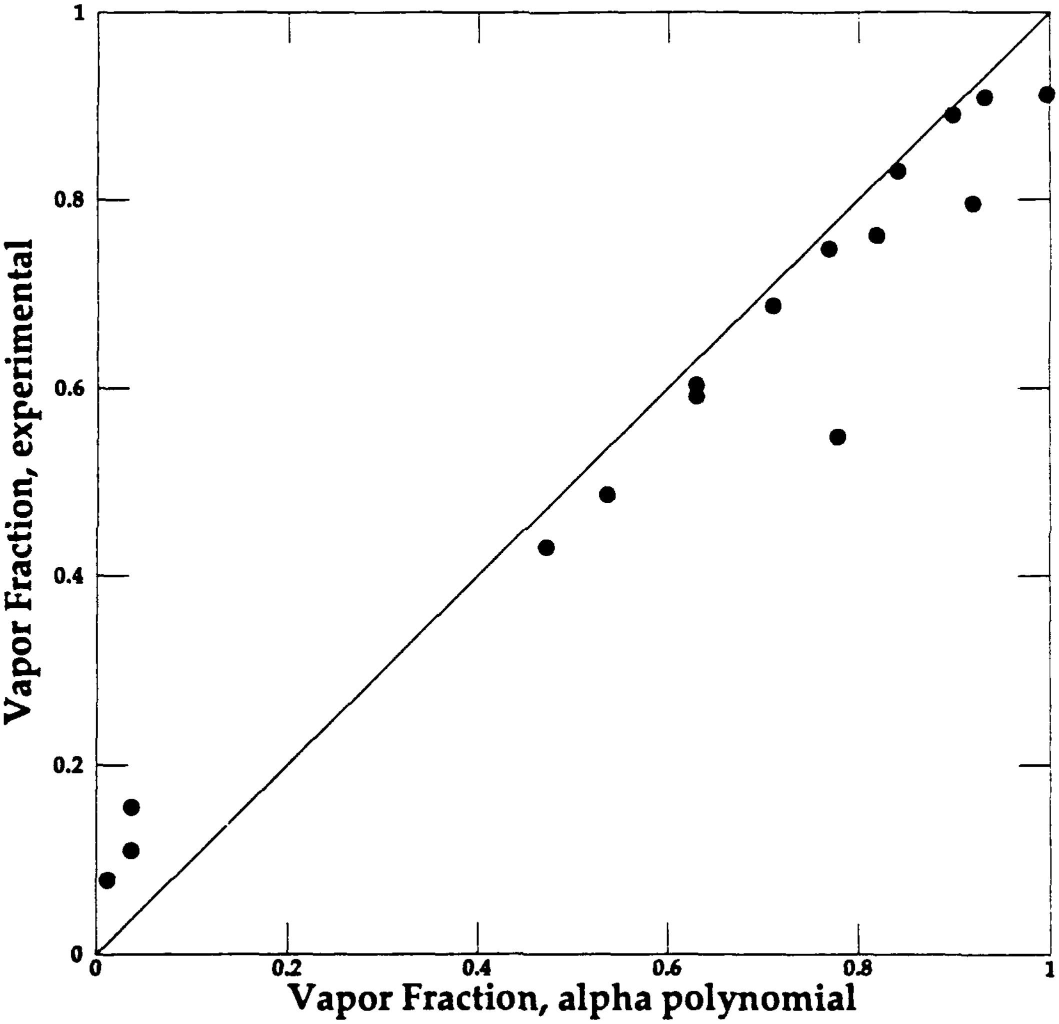


Figure 5.9: A Comparison of Predicted and Experimental Vapor Fractions for a Lumped 4-Component Natural Gas

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

1. The Rachford-Rice objective function can be represented as a polynomial in α, the system vapor fraction. Its coefficients involve elementary symmetric functions, which can be expressed in terms of a determinant whose elements are functions of equilibrium ratios and feed composition.

2. The polynomial has been shown to be well-behaved within the two-phase vapor-liquid region if the system is properly defined in terms of the number of its components. The vapor fraction root on the interval [0,1] can be quickly determined using an ordinary interval-halving technique to provide an initial estimate to the Newton-Raphson iterative method.

3. The regular behavior of the polynomial lends itself to use as a descriptive tool for the conditions of the system within the phase envelope. The Rachford-Rice objective function is not capable of this task as evidenced by Figure 5.7; its unpredictable, singular nature offers no clue to the reason a root was not found on the interval [0,1] for this case. As discussed earlier, a close examination of the curve of the polynomial revealed a local minimum at the experimental value of α. This became a realistic root ($\alpha < 1$) once the system was lumped into four components.

4. The theory of polynomials is well-developed and their behavior and zeros can be predicted with good confidence. By the use of the Fourier-Budan theorem, it can be shown mathematically that only one real root for the α-polynomial can exist on the interval [0,1] for a *well-defined* system. This eliminates the need to solve for all the roots of an N-th order polynomial.

5. The Fourier-Budan theorem can be used as a tool for investigating various lumping schemes whereby multicomponent fluids are condensed to equivalent systems composed of fewer components. The phase behavior of sparse fluids having dilute concentrations of several constituents does not seem to be well-described by the polynomial of degree appropriate to the number of components. In this case, the polynomial yields no roots or at least two roots inside the phase envelope for certain temperature and pressure conditions. It appears that a lumping scheme can be tuned by generating pseudocomponents to give successive polynomials of lower degree until only one root is determined on the interval $0 \leq \alpha \leq 1$.

6.2 Recommendations

1. Further study should focus on coupling the polynomial algorithm to an equation of state and extending this work to flash calculations involving more than two phases.

2. Timing studies could be conducted to determine the exact savings in CPU time realized by the use of the polynomial instead of the Rachford-Rice objective function.

3. Peng et al. (1975) estimate that 75% of the total computing time in compositional reservoir simulation may be related to the phase-behavior part of the program. The savings in computational workload realized by the use of the generalized equation developed in this work might be applied to the employment of an EOS better able to predict fluid thermodynamic properties. The highly nonlinear nature of the equations proposed by Benedict, Webb and Rubin (1940, 1942, 1951) or Lee and Kessler (1975) require iterative solutions but they yield much more accurate representations of fluid behavior, especially of nonhydrocarbon systems.

4. Since the coefficients of the generalized polynomial depend only on the feed composition and equilibrium ratios, research should continue to develop highly accurate K-value prediction methods (e.g., on the basis of convergence pressure). If this can be done with sufficient accuracy, the fugacity-convergence approach and its inherent dependence on an equation of state can be supplanted for flash calculations where nothing more than the phase split and compositions are required. The polynomial algorithm can be solved on a programmable scientific calculator and would provide the engineer with a valuable predictive tool in situations where he or she has no access to a computer capable of running an EOS-based flash routine.

BIBLIOGRAPHY

Abbott, M.M., "Cubic Equations of State: An Interpretative Review," *ACS Advances in Chemistry Series* (1979), **182**, 47-70.

Abhvani, A.S., Beaumont, D.N., "Development of an Efficient Algorithm for the Calculation of Two-Phase Flash Equilibria," *SPE Reservoir Engineering* (Nov 1987), **2**, 695-701.

Asselineau, L., Bogdanic, G., Vidal, J., "A Versatile Algorithm for Calculating Vapour-Liquid Equilibria," *Fluid Phase Equilibria* (Nov 1979), **3**, 273-290.

Baker, L.E., Luks, K.D., "Critical Point and Saturation Pressure Calculations for Multipoint Systems," *SPE Journal* (Feb 1980), **20**, 15-24.

Barbeau, E.J., Polynomials, Springer-Verlag, New York, 1989.

Benedict, M., Webb, G.B., Rubin, L.C., "An Empirical Equation for Thermodynamic Properties of Light Hydrocarbons and Their Mixtures," *Journal of Chemical Physics* (Apr 1940), **8**, 334-345; (Dec 1942), **9**, 747-758.

Benedict, M., Webb, G.B., Rubin, L.C., "An Empirical Equation for Thermodynamic Properties of Light Hydrocarbons and Their Mixtures: Fugacities and Liquid-Vapor Equilibria," *Chemical Engineering Progress* (Aug 1951), **47**, 419; (Sep 1951), **47**, 449-454.

Benmekki, E.-H., "Prediction of the Entire Range of Vapor-Liquid Equilibrium States," M.S. thesis, Illinois Institute of Technology, Chicago, IL, August 1984.

Bloomer, O.T., Gami, D.C., Parent, J.D., "Physical-Chemical Properties of Methane-Ethane Mixtures," Research Bulletin 22, Institute of Gas Technology, Illinois Institute of Technology, Chicago, IL, July 1953.

Boston, J.F., Britt, H.I., "A Radically Different Formulation and Solution of the Single-Stage Flash Problem," *Computers and Chem. Eng.* (1978), **2**, 109-122.

Brinkman, F.H., Sicking, J.N., "Equilibrium Ratios for Reservoir Studies," *Petroleum Transactions, AIME* (1960), **219**, 313-319.

Cox, E.R., "Pressure-Temperature Chart for Hydrocarbon Vapors," *Ind. Eng. Chem.* (Jun 1923), **15**, 592-593.

David, F.N., Kendall, M.G., Barton, D.E., Symmetric Function and Allied Tables, Cambridge University Press, Cambridge, England, 1966.

Dodson, C.R., Standing, M.B.,"Prediction of Volumetric and Phase Behavior of Naturally Occurring Hydrocarbon Systems," *API Drilling and Production Practice* (1941), 326.

Edmister, W.C., "Applications of Thermodynamics to Hydrocarbon Processing: Part XXII--Convergence Correction to Vapor-Liquid Equilibrium Ratios," *Petroleum Refiner* (Sep 1949), **28**, 95-102.

Enick, R.M., Holder, G.D., Mohamed, R., "Four-Phase Flash Equilibrium Calculations Using the Peng-Robinson Equation of State and a Mixing Rule for Asymmetric Systems," *SPE Reservoir Engineering* (Nov 1987), **2**, 687-694.

Fussell, D.D., Yanosik, J.L., "An Iterative Sequence for Phase-Equilibria Calculation Incorporating the Soave-Redlich-Kwong Equation of State," *SPE Journal* (Jun 1978), **18**, 173-182.

Fussell, L.T., "A Technique for Calculating Multiphase Equilibria," *SPE Journal* (Aug 1979), **19**, 203-210.

Gosset, R., Heyen, G., Kalitventzeff, B., "An Efficient Algorithm to Solve Cubic Equations of State," *Fluid Phase Equilibria* (Jan 1986), **25**, 51-64.

Gregory, D.P., Djordjevich, L., Kao, R., Anderson, G.L., "Thermodynamic Properties of a Lean Natural Gas at Cryogenic Conditions," Technical Report 11, Institute of Gas Technology, Illinois Institute of Technology, Chicago, IL, June 1971.

Hadden, S.T., "Vapor-Liquid Equilibria in Hydrocarbon Systems," *Chemical Engineering Progress* (Jan 1948), **44**, 37-54, (Feb 1948), 135-156.

Hadden, S.T., "Convergence Pressure in Hydrocarbon Vapor-Liquid Equilibria," *Chemical Engineering Progress Symposium Series* (1953), **49**, 53-66.

Hanson, G.H., Brown, G.G., "Vapor-Liquid Equilibria in Mixtures of Volatile Paraffins," *Ind. Eng. Chem.* (Sep 1945), **37**, 821-825.

Hoffmann, A.E., Crump, J.S., Hocott, C.R., "Equilibrium Constants for a Gas-Condensate System," *Petroleum Transactions, AIME* (1953), **198**, 1-10.

Huron, M.-J., Vidal, J., "New Mixing Rules in Simple Equations of State for Representing Vapor-Liquid Equilibria of Strongly Non-Ideal Mixtures," *Fluid Phase Equilibria* (1979), **3**, 255-271.

Jhaveri, B.S., Youngren, G.K., "Three-Parameter Modification of the Peng-Robinson Equation of State to Improve Volumetric Predictions," *SPE Reservoir Engineering* (Aug 1988), **3**, 1033-1040.

Katz, D.L., Brown, G.G., "Vapor Pressure and Vaporization of Petroleum Fractions," *Ind. Eng. Chem. Res.* (Dec 1933), **25**, 1373-1384.

Katz, D.L., Hachmuth, K.H., "Vaporization Equilibrium Constants in a Crude Oil-Natural Gas System," *Ind. Eng. Chem. Res.* (Sep 1937), **29**, 1072-1077.

Katz, D.L., Kurata, F., "Retrograde Condensation," *Ind. Eng. Chem. Res.* (Jun 1940), **32**, 817-827.

Kay, W.B., "Density of Hydrocarbon Gases and Vapors," *Ind. Eng. Chem. Res.* (Sep 1936), **28**, 1014-1019.

Knapp, H., Döring, R., Oellrich, L., Plöcker, U., Prausnitz, J.M., Vapor-Liquid Equilibria for Mixtures of Low Boiling Substances, DECHEMA Chemistry Data Series, vol. VI, DECHEMA Deutsche Gesellschaft für Chemisches Apparatewesen, Frankfurt/Main, Federal Republic of Germany, 1982.

Kumar, S., Gas Production Engineering, Gulf Publishing Co., Houston, 1987.

Lawal, A.S., "Consistent Rule for Selecting Roots in Cubic Equations of State," *Ind. Eng. Chem. Res.* (Apr 1987), **26**, 857-859.

Lee, B.I., Kesler, M.G., "A Generalized Thermodynamic Correlation Based on Three-Parameter Corresponding States," *AIChE Journal* (May 1975), **21**, 510-527.

Macdonald, I.G., Symmetric Functions and Hall Polynomials, Oxford University Press, Oxford, England, 1979.

MacMahon, P.A., An Introduction to Combinatory Analysis, Cambridge University Press, London, England, 1920.

Martin, J.J., "Cubic Equations of State--Which?" *Ind. Eng. Chem. Fundam.* (May 1979), **18**, 81-97.

Mathias, P.M., Benson, M.S., "Computational Aspects of Equations of State: Fact and Fiction," *AIChE Journal* (Dec 1986), **32**, 2087-2090.

Mathias, P.M., Copeman, T. W., "Extension of the Peng-Robinson Equation of State to Complex Mixtures: Evaluation of the Various Forms of the Local Composition Concept," *Fluid Phase Equilibria* (Oct 1983), **13**, 91-108.

Mathias, P.M., Naheiri, T., Oh, E.M., "A Density Correction for the Peng-Robinson Equation of State," *Fluid Phase Equilibria* (Jul 1989), **47**, 77-87.

Mehra, R.K., Heidemann, R.A., Aziz, K., "Computation of Multiphase Equilibrium for Compositional Simulation," *SPE Journal* (Feb 1982), **22**, 61-68.

Mehra, R.K., Heidemann, R.A., Aziz, K., "An Accelerated Successive Substitution Algorithm," *Canadian Journal of Chemical Engineering* (Aug 1983), **61**, 590-596.

Michelsen, M.L., "The Isothermal Flash Problem. Part II: Phase-Split Calculation," *Fluid Phase Equilibria* (Sep 1982), **9**, 21-40.

Mott, R.E., "Development and Evaluation of a Method for Calculating Phase Behaviour of Multi-Component Hydrocarbon Mixtures Using an Equation of State," U.K. Atomic Energy Authority, Dorchester (Feb 1980), AEEW-R 1331.

Mott, R.E., "The Winfrith PVT Package," U.K. Atomic Energy Authority, Dorchester (Apr 1983), AEEW-R 1636.

Muskat, M., Physical Principles of Oil Production, McGraw-Hill, New York, 1949.

Nghiem, L.X., Aziz, K., Li, Y.K., "A Robust Iterative Method for Flash Calculations Using the Soave-Redlich-Kwong or the Peng-Robinson Equation of State," *SPE Journal* (Jun 1983), **23**, 521-530.

Parikh, J.S., Bukacek, R.F., Graham, L., Leipziger, S., "Dew and Bubble Point Measurements for a Methane-Ethane-Propane Mixture," *Journal of Chemical and Engineering Data* (1984), **29**, 301-303.

Péneloux, A., Rauzy, E., Fréze, R., "A Consistent Correction for Redlich-Kwong-Soave Volumes," *Fluid Phase Equilibria* (Mar 1982), **8**, 7-23.

Peng, D.-Y., "An Empirical Method for Calculating Vapor-Liquid Critical Points of Multicomponent Mixtures," *Canadian Journal of Chemical Engineering* (Oct 1986), **64**, 827-830.

Peng, D.-Y., Robinson, D.B., "A New Two-Constant Equation of State," *Ind. Eng. Chem. Fundam.* (Feb 1976a), **15**, 59-64.

Peng, D.-Y., Robinson, D.B., "Two and Three Phase Equilibrium Calculations for Systems Containing Water," *Canadian Journal of Chemical Engineering* (Dec 1976b), **54**, 595-599.

Peng, D.-Y., Robinson, D.B., "A Rigorous Method for Predicting the Critical Properties of Multicomponent Systems from an Equation of State," *AIChE Journal* (Mar 1977), **23**, 137-144.

Peng, D.-Y., Robinson, D.B., Bishnoi, P.R., "The Use of the Soave-Redlich-Kwong Equation of State for Predicting Condensate Fluid Behavior," *Proceedings*, Ninth World Petroleum Congress, Tokyo (1975), 377-388.

Pitzer, K.S., Lippmann, D.Z., Curl, R.F. Jr., Huggins, C.M., Petersen, D.E., "The Volumetric and Thermodynamic Properties of Fluids. II. Compressibility Factor, Vapor Pressure and Entropy of Vaporization," *Journal of Amer. Chem. Soc.* (5 Jul 1955), **77**, 3433-3440.

Poling, B., Grens, E.A. II, Prausnitz, J.M., "Thermodynamic Properties from a Cubic Equation of State: Avoiding Trivial Roots and Spurious Derivatives," *Ind. Eng. Chem. Proc. Des. Dev.* (Jan 1981), **20**, 127-130.

Proust, P., Vera, J.H., "PRSV: The Stryjek-Vera Modification of the Peng-Robinson Equation of State. Parameters for Other Pure Compounds of Industrial Interest," *Canadian Journal of Chemical Engineering* (Feb 1989), **67**, 170-173.

Rachford, H.H. Jr., Rice, J.D., "Procedure for Use of Electronic Digital Computers in Calculating Flash Vaporization Hydrocarbon Equilibrium," *Petroleum Transactions, AIME* (1952), **195**, 327-328.

Redlich, O., Kwong, J.N.S., "On the Thermodynamics of Solutions V. An Equation of State. Fugacities of Gaseous Solutions," *Chemical Reviews* (1949), **44**, 233-244.

Reid, R.C., Prausnitz, J.M., Poling, B.E., The Properties of Gases and Liquids, 4th ed., McGraw-Hill, New York, 1987.

Risnes, R., Dalen, V., "Equilibrium Calculations for Coexisting Liquid Phases," *SPE Journal* (Feb 1984), **24**, 87-96.

Risnes, R., Dalen, V., Jensen, J.I., "Phase Equilibrium Calculations in the Near-Critical Region," *Proceedings*, European Symposium on EOR, Bournemouth (1981), 329-350.

Robinson, D.B., "Author's Reply to Comments on 'A New Two-Constant Equation of State'," *Ind. Eng. Chem. Fundam.* (May 1979), **18**, 197.

Robinson, D.B., Peng, D.-Y., "The Characterization of the Heptanes and Heavier Fractions for the GPA Peng-Robinson Programs," Research Report RR-28, Gas Processors Association, Tulsa, OK, March 1978.

Robinson, D.B., Peng, D.-Y., Chung, S.Y.-K., "The Development of the Peng-Robinson Equation and its Application to Phase Equilibrium in a System Containing Methanol," *Fluid Phase Equilibria* (Nov 1985), **24**, 25-41.

Roe, J.R., "Interfunctional Expressibility Problems of Symmetric Functions," Ph.D. dissertation, Syracuse University, Syracuse, NY, June 1918.

Schick, L.M., Prausnitz, J.M., "On the Correlation of Critical Properties of Mixtures," *AIChE Journal* (Jul 1968), **14**, 673.

Soave, G., "Equilibrium Constants from a Modified Redlich-Kwong Equation of State," *Chemical Engineering Science* (Jun 1972), **27**, 1197-1203.

Souders, M. Jr., Selheimer, C.W., Brown, G.G., "Equilibria between Liquid and Vapor Solutions of Paraffin Hydrocarbons," *Ind. Eng. Chem. Res.* (May 1932), **24**, 517-519.

Spencer, C.F., Danner, R.P., "Improved Equation for Prediction of Saturated Liquid Density," *Journal of Chemical and Engineering Data* (1972), **17**, 236-241.

Standing, M.B., Volumetric and Phase Behavior of Oil Field Hydrocarbon Systems, SPE-AIME, Dallas, 1977.

Standing, M.B., "A Set of Equations for Computing Equilibrium Ratios of a Crude Oil/Natural Gas System at Pressures Below 1000 psia," *Journal of Petroleum Technology* (Sep 1979), 1193-1195.

Stryjek, R., Vera, J.H., "An Improved Cubic Equation of State," *American Chemical Society Symposium Series* (1986a), **300**, 561-570.

Stryjek, R., Vera, J.H., "PRSV: An Improved Peng-Robinson Equation of State for Pure Compounds and Mixtures," *Canadian Journal of Chemical Engineering* (Apr 1986b), **64**, 323-333.

Stryjek, R., Vera, J.H., "PRSV--An Improved Peng-Robinson Equation of State with New Mixing Rules for Strongly Nonideal Mixtures," *Canadian Journal of Chemical Engineering* (Apr 1986c), **64**, 334-340.

Stryjek, R., Vera, J.H., "PRSV2: A Cubic Equation of State for Accurate Vapor-Liquid Equilibria Calculations," *Canadian Journal of Chemical Engineering* (Oct 1986d), **64**, 820-826.

van der Waals, J.D., "On the Continuity of the Gaseous and Liquid States," Ph.D. dissertation, Sigthoff University, Leiden, The Netherlands, 1873.

Varotsis, N., "A Robust Prediction Method for Rapid Phase-Behavior Calculations," *SPE Reservoir Engineering* (May 1989), **4**, 237-243.

Varotsis, N., Todd, A.C., Stewart, G., "An Iterative Method for Phase Equilibria Calculations with Particular Application to Multicomponent Miscible Systems," *Proceedings*, European Symposium on EOR, Bournemouth (1981), 313-328.

Vera, J.H., Huron, M.-J., Vidal, J., "On the Flexibility and Limitations of Cubic Equations of State," *Chem. Eng. Comm.* (1984), **26**, 311-318.

Vidal, J., "Equations of State--Reworking the Old Forms," *Fluid Phase Equilibria* (Oct 1983), **13**, 15-33.

Walas, S.M., Phase Equilibria in Chemical Engineering, Butterworth Publishers, Boston, 1985.

Warren, J.H., "Explicit Determination of the Vapor Fraction in Flash Calculations," M.S. thesis, Pennsylvania State University, University Park, PA, December 1991.

White, R.R., Brown, G.G., "Phase Equilibria at High Temperatures," *Ind. Eng. Chem.* (Oct 1942), **34**, 1162-1174.

Whitson, C.H., Torp, S.B., "Evaluating Constant-Volume Depletion Data," paper SPE 10067, *Proceedings*, SPE 56th Annual Fall Technical Conference, San Antonio, TX (5-7 October 1981).

Wilczek-Vera, G., Vera, J.H., "A Comparative Study of Mixing Rules for Cubic Equations of State," *Fluid Phase Equilibria* (1987), **37**, 241-253.

Wilson, G.M., "A Modified Redlich-Kwong Equation of State, Applications to General Physical Data Calculations," paper 15C, *Proceedings*, AIChE National Meeting, Cleveland, OH (4-7 May 1969).

Winn, F.W., "Simplified Nomographic Presentation--Hydrocarbon Vapor-Liquid Equilibria," *Chemical Engineering Progress Symposium Series* (1952), **48**, 121-134.

Wu, D., Sandler, S.I., "Generalized Temperature-Dependent Parameters for the Peng-Robinson Equation of State for *n*-Alkanes," *Ind. Eng. Chem. Res.* (Jul 1989), **28**, 1103-1106.

Xu, Z., Sandler, S.I., "Temperature-Dependent Parameters and the Peng-Robinson Equation of State," *Ind. Eng. Chem. Res.* (Mar 1987a), **26**, 601-606.

Xu, Z., Sandler, S.I., "Application to Mixtures of the Peng-Robinson Equation of State with Fluid-Specific Parameters," *Ind. Eng. Chem. Res.* (Jul 1987b), **26**, 1234-1238.

Zaguskin, V.L., Handbook of Numerical Methods for the Solution of Algebraic and Transcendental Equations, Pergamon Press, London, England, 1961.

Zudkevitch, D., Joffe, J., "Correlation and Prediction of Vapour-Liquid Equilibria with the Redlich-Kwong Equation of State," *AIChE Journal* (1970), **16**, 112-119.

Appendix A

MATHEMATICAL PROOF OF THE GENERALIZED EQUATION

KNOWN:

$$\sum_{i=1}^{N} z_i\, C_i \prod_{j\neq i}^{N} (1+\alpha\, C_j) = \sum_{p=1}^{N} \alpha^{(N-p)} \sum_{i=1}^{N} z_i\, C_i \left\{ a_{N-p}(C_1,\dots,\hat{C}_i,\dots,C_N) \right\} \qquad \text{(A-1)}$$

POSTULATE:

$$\sum_{i=1}^{N} z_i\, C_i \prod_{j\neq i}^{N} (1+\alpha\, C_j) = \sum_{p=1}^{N} \alpha^{(N-p)} \left\{ \prod_{k=1}^{N} C_k \sum_{j=1}^{p} \left[(-1)^{j+1}\, a_{p-j} \sum_{i=1}^{N} \frac{z_i}{C_i^{\,j-1}} \right] \right\} \text{(A-2)}$$

$$= \sum_{p=1}^{N} \alpha^{(N-p)} \sum_{i=1}^{N} z_i C_i \left\{ \prod_{k=1}^{N} C_k \sum_{j=1}^{p} \left[(-1)^{j+1} \frac{a_{p-j}}{C_i^{\,j}} \right] \right\} \text{(A-3)}$$

Prove that the coefficients of α in Equation (A-1) and Equation (A-3) are equivalent:

$$\left\{ a_{N-p}(C_1,\dots,\hat{C}_i,\dots,C_N) \right\} = \left\{ \prod_{k=1}^{N} C_k \left[\sum_{j=1}^{p} (-1)^{j+1} \frac{a_{p-j}}{C_i^{\,j}} \right] \right\} \qquad \text{(A-4)}$$

PROOF:

We can express the *a*-function as:

$$a_{N-p}(C_1,\dots,\hat{C}_i,\dots,C_N) = \frac{\prod_{k=1}^{N} C_k}{C_i}\, a_{p-1}(C_1^{-1},\dots,\hat{C}_i^{-1},\dots,C_N^{-1}) \qquad \text{(A-5)}$$

where $\dfrac{\prod_{k=1}^{N} C_k}{C_i}$ represents $(N-1)$ terms: $N-p = (N-1) - (p-1)$

Eliminate the $\hat{C}_i^{-1}$ term in the RHS of Equation (A-5) by rewriting the *a*-function as:

$$a_{p-1}(C_1^{-1},\dots,\hat{C}_i^{-1},\dots,C_N^{-1}) = a_{p-1}(C_1^{-1},\dots,C_N^{-1}) - C_i^{-1}\, a_{p-2}(C_1^{-1},\dots,C_{N-1}^{-1}) \text{(A-6)}$$

$$a_{p-2}(C_1^{-1},\ldots,\hat{C}_i^{-1},\ldots,C_N^{-1}) = a_{p-2}(C_1^{-1},\ldots,C_N^{-1}) - C_i^{-1}\,a_{p-3}(C_1^{-1},\ldots,C_{N-1}^{-1}) \quad \text{(A-7)}$$

$$a_{p-3}(C_1^{-1},\ldots,\hat{C}_i^{-1},\ldots,C_N^{-1}) = a_{p-3}(C_1^{-1},\ldots,C_N^{-1}) - C_i^{-1}\,a_{p-4}(C_1^{-1},\ldots,C_{N-1}^{-1}) \quad \text{(A-8)}$$

. . . .

$$a_{p-(p-1)}(C_1^{-1},\ldots,\hat{C}_i^{-1},\ldots,C_N^{-1}) = a_{p-(p-1)}(C_1^{-1},\ldots,C_N^{-1}) - C_i^{-(p-1)}\,a_0(C_1^{-1},\ldots,C_{N-1}^{-1}) \quad \text{(A-9)}$$

Recall that $a_0 = 1$ and then substitute Equations (A-7), (A-8),..., (A-9) into Equation (A-6):

$$\begin{aligned} a_{p-1}(C_1^{-1},\ldots,\hat{C}_i^{-1},\ldots,C_N^{-1}) &= a_{p-1}(C_1^{-1},\ldots,C_N^{-1}) - C_i^{-1}\,a_{p-2}(C_1^{-1},\ldots,C_N^{-1}) \\ &+ C_i^{-2}\,a_{p-3}(C_1^{-1},\ldots,C_N^{-1}) - C_i^{-3}\,a_{p-4}(C_1^{-1},\ldots,C_N^{-1}) + \\ &\cdots \pm C_i^{-(p-2)}\,a_{p-(p-1)}(C_1^{-1},\ldots,C_N^{-1}) \pm C_i^{-(p-1)} \end{aligned} \quad \text{(A-10)}$$

After writing the recursive form for the RHS of Equation (A-10), the equation becomes:

$$a_{p-1}(C_1^{-1},\ldots,\hat{C}_i^{-1},\ldots,C_N^{-1}) = \sum_{j=1}^{p} (-1)^{j+1}\,C_i^{-(j-1)}\,a_{p-j} \quad \text{(A-11)}$$

Substitute Equation (A-11) into Equation (A-5):

$$a_{N-p}(C_1,\ldots,\hat{C}_i,\ldots,C_N) = \frac{\prod_{k=1}^{N} C_k}{C_i}\left[\sum_{j=1}^{p} (-1)^{j+1}\,\frac{a_{p-j}}{C_i^{\,j-1}}\right] \quad \text{(A-12)}$$

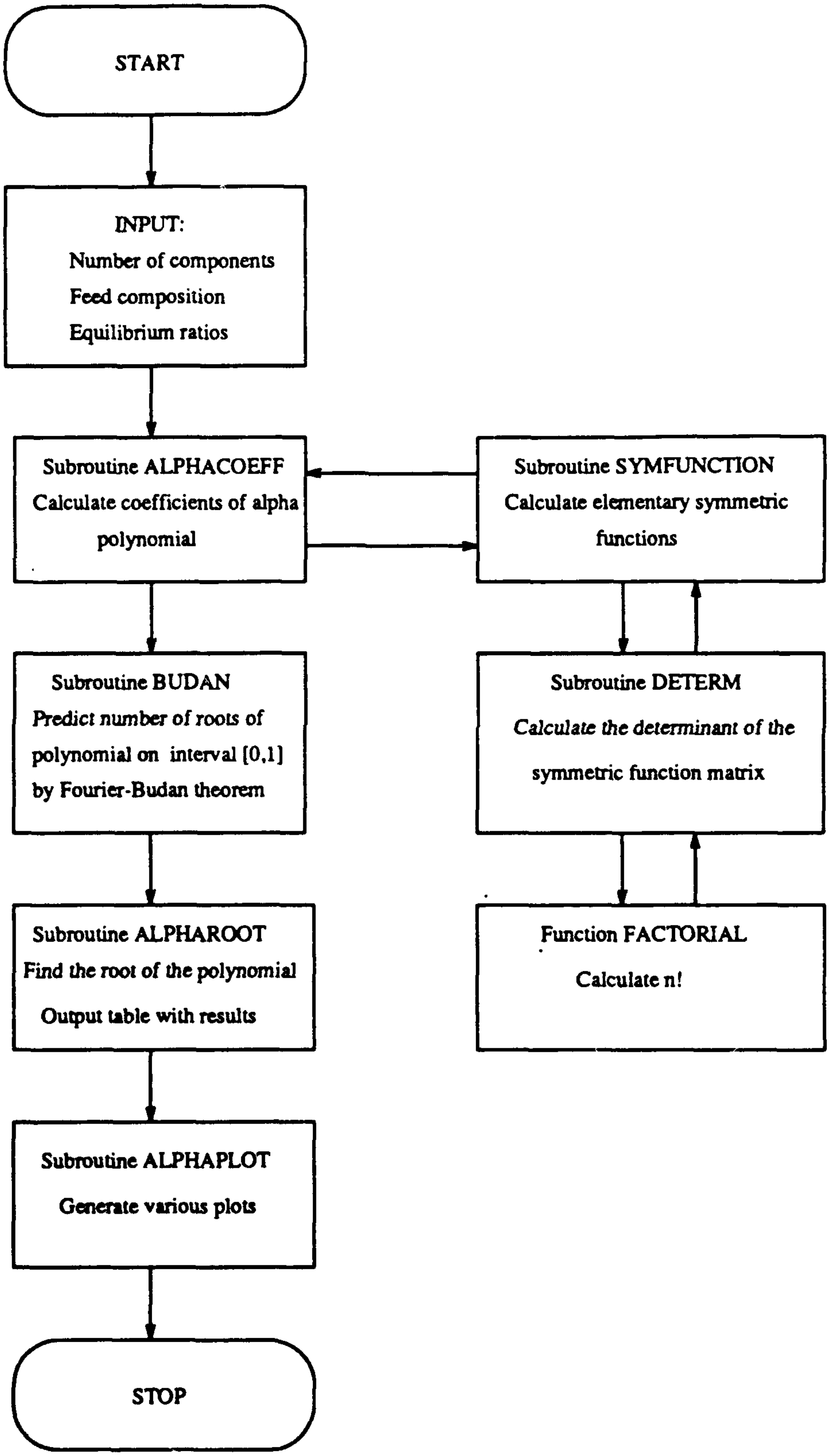
Combine C_i terms:

$$a_{N-p}(C_1,\ldots,\hat{C}_i,\ldots,C_N) = \prod_{k=1}^{N} C_k\left[\sum_{j=1}^{p} (-1)^{j+1}\,\frac{a_{p-j}}{C_i^{\,j}}\right] \quad \text{(A-13)}$$

Equation (A–13) = Equation (A–4) Q.E.D.

Appendix B

ALGORITHM FLOWCHART


START
INPUT:
Number of components
Feed composition
Equilibrium ratios
Subroutine ALPHACOEFF
Calculate coefficients of alpha polynomial
Subroutine SYMFUNCTION
Calculate elementary symmetric functions
Subroutine BUDAN
Predict number of roots of polynomial on interval [0,1] by Fourier-Budan theorem
Subroutine DETERM
Calculate the determinant of the symmetric function matrix
Subroutine ALPHAROOT
Find the root of the polynomial
Output table with results
Function FACTORIAL
Calculate n!
Subroutine ALPHAPLOT
Generate various plots
STOP

Appendix C

COMPUTER CODE

```
*****************************************************************
*
*                               7 Dec 91
*                            BRETT D. WEIGLE
*               Petroleum and Natural Gas Engineering Section
*                      Mineral Engineering Department
*                   College of Earth and Mineral Sciences
*                     The Pennsylvania State University
*                      University Park, Pennsylvania
*
*                               M.S. thesis
*                      Advisor:  Dr. Michael A. Adewumi
*
*****************************************************************
*                      Program ALPHATEST (FORTRAN 77)
*****************************************************************
*
*       This program calculates values of the vapor fraction,
*       given equilibrium ratios, Ki, and feed mole fractions,
*      zi.  It can be used to reproduce experimental results of
*       equilibrium flashes.
*
*       ALPHATEST calls ALPHACOEFF, BUDAN, ALPHAPLOT, and
*                           ALPHAROOT
*       ALPHACOEFF calls subroutine SYMFUNCTION
*       SYMFUNCTION calls subroutine DETERM and function FACTOR
*
*       VARIABLES:  alpha = calculated system vapor fraction
*                    beta = experimental system liquid fraction
*                    coefficient = coefficient of alpha polynomial
*                    Ki = equilibrium ratio for component i
*                    molefrac = feed mole fraction of component i
*                    Ncomp = number of components in feed
*                    Npress = number of data sets to be evaluated
*                    Pi = system pressure, psia
*                    Ti = system temperature, F
*                    xalpha = experimental system vapor fraction
*
*       It is formatted to input zi, temperature, pressure, liquid
*       mole fraction, and Ki
*
*****************************************************************

      IMPLICIT REAL*8(a-h,o-z)
      REAL*8 Ki(500,100),molefrac(0:100)
      PARAMETER(Npress=16,Ncomp=10)
      DIMENSION alpha(500), beta(500), coefficient(0:100),
     @              Pi(500), Ti(500), tarray(2), xalpha(500)
*
*       Data Input
*

*      The number of components (Ncomp) and the number of data sets
*       to be run (Npress) are specified as PARAMETERs'
```

```
*
*          Open and Rewind Input and Output Files
*
      OPEN(unit=1,file='indata',status='old')
      OPEN(unit=7,file='table',status='unknown')
      OPEN(unit=8,file='plot',status='unknown')

      REWIND(unit=1)
      REWIND(unit=7)
      REWIND(unit=8)

      read(1,*) (molefrac(i), i = 1, Ncomp)
      do 1000 j = 1, Npress

         read(1,*) Pi(j), Ti(j), beta(j)
         read(1,*) (Ki(j,i), i = 1, Ncomp)
         xalpha(j) = 1.d0 - beta(j)

 1000 continue
*
*          Choose between single or multiple runs
*
      write(6,*) 'Evaluate one data set? enter 1'
      write(6,*) 'Evaluate all data sets? enter 2'
      read(5,*) numsets

      if(numsets .EQ. 1) then
         write(6,*) 'Enter number of data set for this run'
         read(5,*) j
         go to 2100
      end if

      do 2000 j = 1, Npress

 2100    write(7,*) ' '
         write(7,*) ' '
         write(7,*) '                                    RUN 'j
         write(6,*) 'J = ', j
         write(7,2500) Pi(j),Ti(j),beta(j)
 2500    format('Pressure = ',f6.1,' psia   Temperature = ',f6.1,' F
     @Liquid Mole Fraction = ',f6.4)

*
*          Call subroutines
*
*          Calculate coefficients of polynomial
*
           call ALPHACOEFF(Ncomp,Npress,j,molefrac,Ki,coefficient)

*
*          Predict the number of roots on [0,1] by Fourier-Budan theorem
*
```

```
            call BUDAN(j,Ncomp,coefficient,numroot)
*
*           Solve for the roots by Newton-Raphson method
*
            call ALPHAROOT(j,Ncomp,coefficient,xalpha,numroot,alpha)
*
*           Generate various plots (EDIT the file to remove comments for specific
*           options)
*

            call ALPHAPLOT(Ncomp,j,molefrac,alpha,coefficient,Ki)

*           ALPHAROOT has internal output section to compile a table
*           listing statistics on the determination of alpha

 2000 continue

************************************************************************
*           Produce this format to plot data points as dots:
*                               (PLOTFAT=20)
*
*                               2
*                               x(1) y(1)
*                               x(1) y(1)
*                               2
*                               x(2) y(2)
*                               x(2) y(2)
*                                 etc.
************************************************************************

      do 3000 j = 1, Npress

            write(8,3500) alpha(j),xalpha(j),alpha(j),xalpha(j)
 3500       format('2'/,e16.9,10x,e16.9/,e16.9,10x,e16.9)

 3000 continue

      CLOSE(unit=1)
      CLOSE(unit=7)
      CLOSE(unit=8)

      stop
      end

************************************************************************
*
*                               4 Dec 91
*
*                           BRETT D. WEIGLE
*               Petroleum and Natural Gas Engineering Section
*                      Mineral Engineering Department
*                   College of Earth and Mineral Sciences
*                     The Pennsylvania State University
```

```
*                       University Park, Pennsylvania
*
*                               M.S. thesis
*
*                          SUBROUTINE ALPHACOEFF
*
*        This subroutine calculates the coefficient for each term in
*        the general polynomial for the vapor fraction, alpha:
*
*        P(alpha) = c0 + c1*alpha + c2*alpha**2 + ... +
*
*                                          c(Ncomp-1)*alpha**(Ncomp-1)
*        Equation 4.29 in the thesis.
*
***********************************************************************

      SUBROUTINE ALPHACOEFF(Ncomp,Npress,jj,molefrac,Ki,coefficient)

      IMPLICIT REAL*8(a-h,o-z)
      REAL*8 Ki(500,100), molefrac(0:100)
      INTEGER p

      DIMENSION coefficient(0:100), c(100)

      OPEN(unit=14,file='coeff',status='unknown')
      OPEN(unit=15,file='coeff.plot',status='unknown')

      if(Ncomp .LT. 2) then
         write(6,*)'You cannot flash this system'
         stop
      end if

*
*        Calculate Ci = Ki - 1
*
      do 0500 k = 1, Ncomp
         c(k) = Ki(jj,k) - 1.d00
 0500 continue
*
*        p-loop increments the power of alpha
*

C       write(15,*)Ncomp

      do 1000 p = 1, Ncomp
         temporary = 0.d00

         do 2000 j = 1, p

*        Zero-order elementary symmetric function, a0[1/Ci], defined as 1
*
            if(p-j .EQ. 0) then
               apj = 1.d00
               go to 2500
```

```
                    end if

*                 Call subroutine to calculate the elementary symmetric
*                 function, apj

                    call SYMFUNCTION(Ncomp,j,p,c,apj)

 2500              ratio = 0.d00

                    do 3000 i = 1, Ncomp
                        ratio = ratio + molefrac(i)/(c(i)**(j-1))
 3000              continue

                    temporary = temporary + ((-1.d0)**(j+1))*apj*ratio
 2000          continue

          coefficient(Ncomp-p) = temporary

C           write(14,*)'Coefficient(',Ncomp-p,') = ',coefficient(Ncomp-p)
C           write(15,*)Ncomp-p,coefficient(Ncomp-p)

 1000 continue

          return
          end

***********************************************************************
*
*                                   4 Dec 91
*
*                                BRETT D. WEIGLE
*                  Petroleum and Natural Gas Engineering Section
*                          Mineral Engineering Department
*                       College of Earth and Mineral Sciences
*                          The Pennsylvania State University
*                           University Park, Pennsylvania
*
*                                    M.S. thesis
*
*                              SUBROUTINE SYMFUNCTION
*
*         This subroutine calculates the elementary symmetric function
*
*                                   a(p-j){1/Ci}
*
***********************************************************************

          SUBROUTINE SYMFUNCTION(Ncomp,j,p,c,apj)

          IMPLICIT REAL*8(a-h,o-z)
          REAL*8 mmatrix(100,100)
          INTEGER factor,p

          DIMENSION c(100), s(100)
```

```
*
*          Compute the power-sum series:  s = sigma[ (1/Ci)**lambda ]
*

      n = p - j
      do 1000 lambda = 1, n
         sum = 0.d00

         do 2000 i = 1, Ncomp
            sum = sum + (1.d0/c(i))**lambda
 2000    continue
         s(lambda) = sum
 1000 continue

*          Build the matrix MMATRIX

      do 3000 k = 1, n

         do 4000 l = 1, n
            if(l .LE. k) mmatrix(k,l) = s(k-l+1)
            if(l .EQ. k+1) mmatrix(k,l) = DFLOAT(k)
            if(l .GT. k+1) mmatrix(k,l) = 0.d00
 4000    continue

 3000 continue

*          Since a1{1/Ci} forms a [1x1] matrix, its determinant is the
*          element itself

      if(p-j .EQ. 1) then
        det = mmatrix(1,1)
        go to 5000
      end if
*
*          Compute the determinant of MMATRIX
*

      call DETERM(mmatrix,n,det)

*
*          Compute the elementary symmetric function
*

 5000 apj = det/factor(n)

      return
      end

*********************************************************************
*
*                    Function to compute the factorial
*
*********************************************************************
```

```
      FUNCTION factor(n)
      INTEGER factor,i,n

      factor = 1
      if(n .GT. 0) then
         do 6000 i = 2,n
            factor = factor*i
 6000    continue
      end if
      end

*********************************************************************
*
*                           4 Dec 91
*
*                        BRETT D. WEIGLE
*            Petroleum and Natural Gas Engineering Section
*                  Mineral Engineering Department
*               College of Earth and Mineral Sciences
*                 The Pennsylvania State University
*                   University Park, Pennsylvania
*
*                           M.S. thesis
*
*                        SUBROUTINE DETERM
*
*  This program calculates the determinant of an NxN matrix.
*  First, partial pivoting is performed on a nonsingular matrix by
*  Gaussian elimination.  This produces a triangular matrix whose
*  determinant can be calculated by computing the product of all
*  the diagonal entries.
*  The augmented matrix does not contain the normal last column
*  which represents the right-hand side of a system of linear
*  equations; AUG is the same as the original matrix.
*
*  VARIABLES:
*            N = dimension of matrix
*            AUG = augmented matrix
*            I,J,K = indices
*            MULT = multiplier used to eliminate an unknown
*            PIVOT = used to find nonzero diagonal entry
*********************************************************************

      SUBROUTINE DETERM(aug,n,det)

      IMPLICIT REAL*8(a-h,o-z)
      REAL*8 mult
      INTEGER pivot
      DIMENSION aug(100,100)
*
*        Gaussian elimination
*
```

```
      do 7000 i = 1, n
*
*          Locate nonzero entry
*
         if(aug(i,i) .EQ. 0) then
           pivot = 0
           j = i + 1
 3000     if((pivot .EQ. 0) .AND. (j .LE. n)) then
              if(aug(j,i) .NE. 0) pivot = j
              j = j + 1
              go to 3000
           end if

           if(pivot .EQ. 0) then
              print *,'Matrix is singular'
              stop
           else
*
*          Interchange rows I and PIVOT
*
           do 4000 j = i, n
               temp = aug(i,j)
               aug(i,j) = aug(pivot,j)
               aug(pivot,j) = temp
 4000     continue

           end if

         end if
*
*          Eliminate I-th unknown from equations I+1, ..., N
*
         do 6000 j = i+1, n
            mult = -aug(j,i) / aug(i,i)

            do 5000 k = i, n
               aug(j,k) = aug(j,k) + mult * aug(i,k)
 5000       continue

 6000   continue

 7000 continue

*          Calculate the determinant of matrix AUG by computing the
*          product of the diagonal elements

      prod = 1.d0
      do 8000 i = 1, n
```

```
          do 9000 j = 1, n
             if(i .EQ. j) prod = prod * aug(i,j)
 9000     continue

 8000 continue

      det = prod

      return
      end

***********************************************************************
*
*                               4 Dec 91
*
*                            BRETT D. WEIGLE
*                 Petroleum and Natural Gas Engineering Section
*                       Mineral Engineering Department
*                     College of Earth and Mineral Sciences
*                       The Pennsylvania State University
*                        University Park, Pennsylvania
*
*                               M.S. thesis
*
*                          SUBROUTINE ALPHAROOT
*
*         Subroutine uses an interval-halving technique to find
*         the best root value to initialize the Newton-Raphson (N-R)
*         iterative calculations which determine the real root of
*         the alpha polynomial on the interval [0,1].
*
*         PARAMETERS: delta = alpha increment
*                     epsilon = alpha convergence criterion
*         VARIABLES:  alower = lower bound of alpha increment
*                      aupper = upper bound of alpha increment
*                      falpha = the alpha polynomial
*                      fprime = first derivative of alpha polynomial
*                      guess = iterative variable for alpha
*                      guess0 = initial estimate for N-R
*                      intcount = # of intervals until sign change
*                      iter = # of iterations until N-R converged
*                      isign,isign2 = flags for function sign change
*                      isign,isign2 = flags for function sign change
*                      numroot = flag for # of zeros (from BUDAN)
***********************************************************************

      SUBROUTINE ALPHAROOT(j,Ncomp,coefficient,xalpha,numroot,
     @                      alpha)

      IMPLICIT REAL*8(a-h,o-z)
      INTEGER p

      DIMENSION alpha(500), coefficient(0:100), xalpha(500)
```

```
      PARAMETER(delta = 0.01d0, epsilon = 1.d–06)


*
*         Write table heading
*

      write(7,*)'The Fourier–Budan Theorem yields ',numroot,' roots on
     @this interval'
      write(7,3500)
 3500 format('Intervals',4x,'Initial Guess',4x,'Iterations',4x,'Calc.
     @Alpha',4x,'Exp. Alpha')

*         Check flag NUMROOT provided by subroutine BUDAN to determine
*         root–search scheme

      if(numroot .EQ. 0) then
         write(6,*) 'No root on the interval [0,1] for data set ',j
         intcount = 0
         write(7,3900) intcount,xalpha(j)
 3900    format(i4,61x,f5.3)
         write(7,*)'No root on the interval [0,1]'
         return
      end if

      if(numroot .EQ. 1) then
         ilower = 0
         iupper = 0
      end if

      if(numroot .GE. 2) then
         ilower = 0
         iupper = 1
      end if

*         Use incremental search to determine initial guess
*                   Interval Endpoint DO–Loop

      do 0400 jroot = ilower, iupper

         intcount = 0
*
*        Test the polynomial at endpoint for initial sign value
*

         if(jroot. EQ. ilower) then
            guess = DFLOAT(ilower)
            alower = guess
            aupper = alower + delta
         end if

         if(jroot. EQ. iupper) then
            guess = DFLOAT(iupper)
```

```
            aupper = guess
            alower = aupper - delta
         end if
         ichange = 0

 0600    falpha = 0.d0
         do 1500 p = 1, Ncomp

             term = coefficient(Ncomp-p)*guess**(Ncomp-p)
             if( (Ncomp-p) .EQ. 0 ) term = coefficient(0)
             falpha = falpha + term

 1500    continue
*
*        Initialize ISIGN2 on first pass with endpoint
*

         if(ichange .EQ. 0) then
            if(falpha .GE. 0.) then
               isign2 = 1
            else
               isign2 = 0
            end if
         end if

*
*        Note the sign of the function
*

         if(falpha .GE. 0.) then
            isign = 1
         else
            isign = 0
         end if

*        Test function for sign change and increment or decrement the
*        search variable as appropriate

         if(isign2 .EQ. isign) then
            if(jroot .EQ. ilower) then
               alower = aupper
               aupper = aupper + delta
               guess = aupper
            else if(jroot .EQ. iupper) then
               aupper = alower
               alower = aupper - delta
               guess = alower
            end if
         end if

*
*        Exit subroutine if no sign change is detected on interval [0,1]
*
```

```
          if( (guess .GT. 1.) .OR. (guess .LT. 0.) ) then
             write(6,*) 'No root on the interval [0,1]'
             write(7,3800) intcount,xalpha(j)
3800        format(i4,61x,f5.3)
             write(7,*)'No root on the interval [0,1]'
             return
          end if

*
*         If NO sign change but still within interval, repeat the sequence
*
          if(isign .EQ. isign2) then
             isign2 = isign
             intcount = intcount + 1
             ichange = 1
             go to 0600

          else

*         If there IS a sign change:
*         Halve the interval where the function crosses the x axis

             guess0 = (alower + aupper) / 2.d0
          end if

*
*         Provide this guess to Newton-Raphson to begin calculations
*

          guess = guess0

*
*         N-R is limited to 1000 iterations for convergence
*

          iter = 0

          do 1000 iterlimit = 1, 1000

              iter = iter + 1
              falpha = 0.d00
              fprime = 0.d00

              do 2000 p = 1, Ncomp
                  falpha = falpha + coefficient(Ncomp-p)
     @                    *guess**(Ncomp-p)
                  fprime = fprime + (Ncomp-p)*coefficient(Ncomp-p)
     @                    *guess**(Ncomp-p-1)
 2000          continue

              calc = guess - falpha/fprime
              error = DABS((calc - guess)/calc)
              guess = calc
              if(error .LE. epsilon) go to 3000
```

```
 1000      continue

        print *,'N-R method failed to converge after 1000 iterations'

*
*          Output results to file "TABLE"
*

 3000      write(7,3600) intcount,guess0,iter,guess,xalpha(j)
 3600      format(i4,13x,f5.3,10x,i4,13x,f9.6,7x,f5.3)

           alpha(j) = guess
*
*          Begin search for root from opposite end of interval
*

 0400 continue

        return
        end

************************************************************************
*
*                               6 Dec 91
*
*                            BRETT D. WEIGLE
*                 Petroleum and Natural Gas Engineering Section
*                      Mineral Engineering Department
*                    College of Earth and Mineral Sciences
*                      The Pennsylvania State University
*                       University Park, Pennsylvania
*
*                               M.S. thesis
*
*                          SUBROUTINE ALPHAPLOT
*
*          This subroutine is used for several purposes:
*          1. Plotting F(alpha) vs alpha [Rachford-Rice obj function]
*          2. Plotting F(alpha) vs alpha [polynomial]
*          3. Plotting Fprime vs alpha [polynomial]
*
************************************************************************

        SUBROUTINE ALPHAPLOT(Ncomp,J,molefrac,alpha,coefficient,Ki)

        IMPLICIT REAL*8 (a-h,o-z)
        REAL*8 Ki(500,100),molefrac(100)
        DIMENSION alpha(500),coefficient(0:100)

        INTEGER p

        PARAMETER(start = 0.0d0, end = 2.0d0, stepsize = 0.0005d0)

        OPEN(unit=11,file='fa.plot',status='unknown')
```

```
      OPEN(unit=12,file='fprime.plot',status='unknown')

*
*        Number of data points for plotting
*

      number = IDINT((end - start + stepsize)/stepsize)

*******************************************************************
*          F(alpha) vs alpha [polynomial]
*          F'(alpha) vs alpha [polynomial]
*                  Adjust Ncomp,Npress in PARAMETER

      write(11,*) number
      do 1000 phase = start,end,stepsize
         falpha = 0.d00
         fprime = 0.d00
         do 2000 p = 1,Ncomp
            falpha = falpha + coefficient(Ncomp-p)*
     @                        phase**(Ncomp-p)
C           fprime = fprime + (Ncomp-p)*coefficient(Ncomp-p)*
C    @                        phase**(Ncomp-p-1)
 2000    continue
         write(11,3600) phase,falpha
C        write(11,3600) phase,fprime
 3600    format(f7.3,2x,f25.12)
 1000 continue
*******************************************************************
C*                 Rachford-Rice objective function
C
C      do 4500 k = 1,Npress
C      k = 6
C      write(11,*) number
C         do 3000 phase = start,end,stepsize
C            falpha = 0.d00
C            do 4000 i = 1,Ncomp
C               falpha = falpha + (molefrac(i)*(Ki(k,i) - 1.d0)) /
C    @                       (1.d00 + phase*(Ki(k,i) - 1.d0))
C*                 End of i loop
C 4000       continue
C
C            write(11,3500) phase,falpha
C 3500       format(f7.3,2x,f25.12)
C*                 End of phase loop
C 3000    continue
C*                 End of k loop
C 4500 continue
*******************************************************************

      CLOSE(unit=11)
      CLOSE(unit=12)

      return
      end
```

```
*********************************************************************
*
*                          6 Dec 91
*
*                       BRETT D. WEIGLE
*          Petroleum and Natural Gas Engineering Section
*                Mineral Engineering Department
*             College of Earth and Mineral Sciences
*               The Pennsylvania State University
*                 University Park, Pennsylvania
*
*                          M.S. thesis
*
*                       SUBROUTINE BUDAN
*
*        Subroutine uses the Fourier-Budan Theorem to determine
*        the number of roots that the alpha polynomial has on the
*       interval [u,v].
*
*        PARAMETERS: iu = lower bound of alpha interval
*                    iv = uppper bound of alpha interval
*        VARIABLES:  coefficient = coefficient of alpha polynomial
*                    dcoeff = coefficient of polynomial derivatives
*                    deriv = derivatives of alpha polynomial
*                    fvapor = the alpha polynomial
*                    ia,ib = # of sign changes for derivative series
*                    ivapor = alpha = vapor fraction
*                    jsign,ksign = flags for derivative sign change
*                    numroot = number of zeros on the interval
*********************************************************************

      SUBROUTINE BUDAN(J,Ncomp,coefficient,numroot)

      IMPLICIT REAL*8(a-h,o-z)
      INTEGER p

      DIMENSION dcoeff(0:100,0:100), coefficient(0:100), deriv(0:100)

      PARAMETER(iu = 0, iv = 1)


C      DATA (coefficient(l), l = 0,Ncomp-1) /-1.,1.,-2.,3.,-4.,5. /
      OPEN(unit=2,file='test',status='unknown')
      REWIND(unit=2)

      ia = 0
      ib = 0
      do 0500 ivapor = iu, iv, 1
*
*        Evaluate the polynomial function at the endpoints iu and iv
*

         fvapor = 0.d0
```

```
          do 0600 p = 1, Ncomp
              fvapor = fvapor + coefficient(Ncomp-p)*ivapor**(Ncomp-p)
 0600     continue
                  write(2,*) ´fvapor = ´,fvapor
                  write(2,*) ´ ´

*
*         Calculate coefficients of first derivative
*


          do 1000 n = Ncomp-1, 0, -1
              dcoeff(0,n) = coefficient(n)
              write (2,*) ´dcoeff(0,´,n,´) = ´,dcoeff(0,n)
 1000     continue
                  write(2,*) ´ ´

*         Calculate coefficients of 2nd- and higher-order derivatives
*         as multiples of those of the first derivative

          do 1500 m = 1, Ncomp-1

              do 2000 n = Ncomp-m, 1, -1
                  dcoeff(m,n-1) = n*dcoeff(m-1,n)
              write (2,*) ´dcoeff(´,m,´,´,n-1,´) = ´,
      @dcoeff(m,n-1)
 2000         continue
                  write(2,*) ´ ´

 1500     continue

*
*         Evaluate the derivative series at the endpoints iu and iv
*

          do 3000 m = 1, Ncomp-1
              deriv(m) = 0.d0

              do 4000 n = Ncomp-m, 1, -1
                  term = dcoeff(m,n-1)*ivapor**(n-1)
                  if( (n-1) .EQ. 0 ) term = dcoeff(m,n-1)
                  deriv(m) = deriv(m) + term
                  write(2,*) ´inter deriv(´,m,´) = ´,deriv(m)
 4000         continue

                  write(2,*) ´total deriv(´,m,´) = ´,deriv(m)
                  write(2,*) ´ ´

 3000     continue

*
*         Count the sign changes between the terms of the series
*

          if(fvapor. LT. 0.) then
```

```
           ksign = 0
         else
           ksign = 1
         end if
               write(2,*) 'ksign = ',ksign,' for fvapor'

         do 5000 i = 1, Ncomp-1
            if(deriv(i) .LT. 0.) then
              jsign = 0
            else
              jsign = 1
            end if
               write(2,*) 'jsign = ',jsign,' for deriv(',i,')'

*
*        Increment A or B, depending upon the endpoint under evaluation
*

            if(ivapor .EQ. iu) then
              if(ksign .NE. jsign) then
                ia = ia + 1
                write(2,*) 'ia = ',ia,' for deriv(',i,')'
              end if
            end if

            if(ivapor .EQ. iv) then
              if(ksign .NE. jsign) then
                ib = ib + 1
                write(2,*) 'ib = ',ib,' for deriv(',i,')'
              end if
            end if

            ksign = jsign
              write(2,*) 'ksign = ',ksign,' after deriv(',i,')'
              write(2,*) ' '
 5000       continue

 0500 continue

*
*        Pass a flag to calling program to indicate root conditions
*

      write(2,*) 'ia = ',ia,' and ib = ',ib
      numroot = ia - ib
      write(2,*) 'numroot = ',numroot

      write(2,6000) Ncomp-1, numroot, iu, iv, J
 6000 format('This polynomial of order ',i3,' has ',i3,' zeros on the in
     @terval [',i2,',',i2,'] for J = ',i3)

      CLOSE(unit=2)
      return
      end
```