ATMIRE

Title	Representation of thermodynamic properties by local approximation
Item Type	Text
Authors	Musallam, Waleed
Publisher	Colorado School of Mines. Arthur Lakes Library
Rights	Copyright of the original work is retained by the author.
Download date	2026-11-27 21:17:46
Link to Item	https://hdl.handle.net/11124/13519

T-3612

Representation of Thermodynamic Properties by Local Approximations

by

Waleed Musallam

ProQuest Number: 10783044

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.



ProQuest 10783044

Published by ProQuest LLC (2018). Copyright of the Dissertation is held by the Author.

All rights reserved.
This work is protected against unauthorized copying under Title 17, United States Code
Microform Edition © ProQuest LLC.

ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 – 1346

T-3612

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Chemical and Petroleum-Refining Engineering).

Golden, Colorado
Date _____

Signed: _____

Waleed Musallam

Approved: _____

Paul F. Bryan
Thesis Advisor

Golden, Colorado
Date 11/23/88

_____

Arthur J. Kidnay, Head
Department of Chemical
Engineering and Petroleum
Refining

Representation of thermodynamic properties by Local Approximations

Abstract

In the course of non-ideal phase-equilibrium calculations, thermodynamic properties are often obtained from complex molecular thermodynamic (MT) correlations. Since these calculations require iterative schemes, the use of such expessions can become prohibitively expensive. In this work, local-approximation (LA) models have been used to replace MT models for many of the required thermodynamic evaluations in single-stage liquid-liquid and vapor-liquid equilibrium calculations. The LA models are simple analytic forms that represent activity coefficients and their composition derivatives over a small region of thermodynamic state space occupied by a single equilibrium unit.

Pseudo-ternary and pseudo-quartenary two-suffix Margules equations give good representation of activity coefficient behavior and require only three and six parameters, respectively, per component. Taylor-series expansions of these models around composition base points close to the solution are also investigated. The parameters of these models are evaluated by regression from approximately twice

the minimum number of data points needed to define the parameters. The influence of the algebraic form of the LA model and of the criteria for the selection of regression data points have been investigated.

LA models for each component in each liquid phase are created at the outset of the iterative convergence procedure. Once the iteration has converged to a result **consistent with the local approximation**, one of two procedures is undertaken:

1)The iterations are continued with thermodynamic properties obtained from the MT model (in order to arrive at the 'exact' solution), but with composition derivatives from the LA model.

2)In a 'moving base point' strategy, the Taylor-series expansions are modified by replacing the old base points with the 'first' converged solution. This is repeated until the 'exact' solution is achieved.

By replacing advanced MT models with much simpler analytic forms, the computational costs involved with obtaining activity coefficients and compositional derivatives of these activity coefficients can be substantially reduced.

TABLE OF CONTENTS

I - Introduction 1

II - Mathematical development 25

III - Results and discussion of single-stage flash 60

IV - Conclusion 108

Bibliography 110

APPENDICES

Appendix A 114

Appendix B 116

Appendix C 118

Appendix D 125

Appendix E 131

Appendix F 134

Appendix G 137

Appendix H 139

Appendix I 142

Appendix J 145

Appendix K 151

Appendix L 154

LIST OF FIGURES

page

Figure 1 Schematic of A Single Equilibrium Stage..... 26

Figure 2 Phase Diagram for Ternary System 6..... 64

Figure 3 Phase Diagrams for The Partially Miscible Ternary Sub-systems of System 8..... 65

Figure 4 Phase Diagrams of Partially Miscible Ternary Sub-systems of System 9..... 66

Figure 5 Phase Diagrams for Partially Miscible Ternary Sub-systems of System 10..... 67

LIST OF TABLES

page

Table 1 Multicomponent Systems Investigated..... 61

Table 2 Results of VLE Calculations by Successive -Substitution for Ternary System Acetone-Methanol-Water..... 69

Table 3 Results of VLE Calculations by Successive -Substitution for Ternary System Hexane-Benzene-Ethanol..... 70

Table 4 Results of VLE Calculations by Successive -Substitution for quarternary System Methanol-Ethanol-Acetone-Water..... 71

Table 5 Results of VLE Calculations by Successive -Substitution for quarternary System Cyclohexane-Benzene-Toluene-Ethanol..... 72

Table 6 Results of VLE Calculations by Successive -Substitution for Six Component System Hexane-Heptane-Cyclohexane-Benzene-Toluene-Ethanol..... 73

Table 7 Results of LLE Calculations by Successive -Substitution for Ternary System Acetonitrile-Methanol-n-octane..... 74

Table 8 Results of LLE Calculations by Successive -Substitution for Ternary System DMF-Water-Benzene..... 75

Table 9 Results of LLE Calculations by Successive -Substitution for quarternary System Heptane-DMF-Water-Benzene..... 76

Table 10 Results of LLE Calculations by Successive -Substitution for quarternary System Iso-octane-Cyclohexane-Benzene-Toluene..... 77

Table 11 Results of LLE Calculations by Successive -Substitution for Six Component System Benzene-Toluene-Cyclohexane-Hexane-Heptane-Sulfolane..... 78

Table 12 Costs in Terms of MT Evaluations for the NR Method (VLE mixtures)..... 80

Table 13 Costs in Terms of MT Evaluations for the NR Method (LLE mixtures)..... 81

Table 14 Costs in Terms of MT Evaluations for the LA1 Method (VLE mixtures)..... 88

Table 15 Costs in Terms of MT Evaluations for the LA1 Method (LLE mixtures)..... 89

Table 16 Costs in Terms of MT Evaluations for the LA2 Method (VLE mixtures)..... 90

Table 17 Costs in Terms of MT Evaluations for the LA2 Method (LLE mixtures)..... 91

Table 18 Costs in Terms of MT Evaluations for the LA3 Method (VLE mixtures)..... 92

Table 19 Costs in Terms of MT Evaluations for the LA3 Method (LLE mixtures)..... 93

Table 20 Costs in Terms of MT Evaluations for the LA5a Method (VLE mixtures)..... 95

Table 21 Costs in Terms of MT Evaluations for the LA5a Method (LLE mixtures)..... 96

Table 22 Costs in Terms of MT Evaluations for the LA6a Method (VLE mixtures)..... 97

Table 23 Costs in Terms of MT Evaluations for the LA6a Method (LLE mixtures)..... 98

Table 24 Computational Efficiencies of The SS, NR, and LA Convergence Methods (VLE)..... 99

Table 25 Computational Efficiencies of The SS, NR, and LA Convergence Methods (LLE).....100

Table 26 Costs in Terms of MT Evaluations for the LA5b Method (VLE mixtures)..... 102

Table 27 Costs in Terms of MT Evaluations for the LA5b Method (LLE mixtures).....103

Table 28 Costs in Terms of MT Evaluations for the LA6b Method (VLE mixtures).....104

Table 29 Costs in Terms of MT Evaluations for the LA6b Method (LLE mixtures).....105

Table 30 Computational Efficiencies of The LA5a,LA5b, LA6a, LA6b Models (VLE mixtures).....106

Table 31 Computational Efficiencies of The LA5a,LA5b, LA6a, LA6b Models (LLE mixtures).....107

I-INTRODUCTION

The separation of mixtures into pure components has been a human endeavor since antiquity. Where once it produced the luxuries of body oils, perfumes and alcohol, today, it provides many of the necessities of modern life i.e. transportation fuels and pharmaceuticals. It is frequently used in chemical processing, and it is not unusual to find much of the capital investment in a chemical plant attributed to separation processes e.g. King(1981) reports that separation equipment accounts for up to 90% of the capital investment in large scale petroleum and petrochemical processes.

Separation of the constituents of a homogeneous mixture is usually performed by creating or introducing a second phase, where, due to uneven distribution of the components between the two phases, a separation occurs. In a distillation process, energy is added to a liquid mixture to cause a vapor phase to form. The more volatile components will tend to concentrate in the vapor phase, and the less volatile components in the liquid phase. Physical separation of the phases is then affected by gravity. Distillation will not always be feasible due to certain limitations (Brian, 1972) : 1) The operation temperature must be between the

triple-point and the critical-point temperatures for the components being separated. This temperature range might be too hot or too cold for certain processes. 2) The vapor pressures of the components will control the pressure of the distillaton operation. If the components have low volatilities, the pressure level will be low unless high operating tempertures are used. If, in addition, these components are unstable at high tempertures, costly high vacuum conditions must be created (the operating temperature is then kept low). Therefore, for nonvolatile and/or heat sensitive materials, an alternative to distillation must be considered.

In liquid-extraction processes, where a new phase is created by a mass-separation agent, a second liquid solvent is added to a homogeneous liquid solution. This solvent is immiscible or partially miscible with the solution. At equilibrium, the components will distribute unevenly between the two liquid phases formed, and a separation will occur due to the solutes interacting differently with the solvent.

Phase equilibrium calculations are always needed in process simulation and design. When the systems considered are highly non-ideal, complex and computationally expensive, molecular-thermodynamic (MT) models are needed to provide

equilibrium properties. Prausnitz (Prausnitz et al.,1980) reports that in many calculations, up to 95% of the computational effort is involved with these property evaluations. Tyreus et al.(1976), studying the dynamics of a 40-tray binary distillation column, found that on the order of 400,000 iterative bubble-point calculations were required. Thus, a multiple column configuration with non-ideal mixtures could require millions of calculations. In addition, design studies utilizing process simulations and optimizations require repeated 'stream' flashes in order to determine the phase conditions of all of the process streams encountered. Clearly, more efficient computational methods are required in order to significantly reduce the number of thermodynamic evaluations required in more traditional methods. One approach to this problem is to reduce the computational costs of the property evaluations without affecting the accuracy of the results. Another is to reduce the number of property evaluations needed for a given calculation.

One way to approach this problem is though the use of local approximations. In the local approximation (LA) method, complex thermodynamic behavior is modeled by simple functional expressions (aptly called local models) fit to an MT model over limited regions of composition, temperature,

and pressure. As the regions of interest change during the iterative calculations, parameters in the LA models can be periodically updated using rigorous thermodynamic properties from the underlying MT model. This ensures that the LA model will always be appropriate to the current local region of interest.

By replacing a complex MT model by an LA model for most of the iterative calculation procedure, accurate representation of phase equilibia will be maintained at a cost less than by the use of rigorous MT evaluations alone. Of course, the application of such a model depends on the selection of the form of the model, the determination of the updating criteria, and the model's implementation and termination in the calculation procedure. Before the efforts of this work are presented, a brief introduction to single-stage equilibrium calculations and a description of previous LA work will be given.

The Single-stage Isothermal Flash Calculation

In the chemical process industries, the calculation of equilibrium phase compositions is an important task. The single-stage isothermal-flash is a representative problem that requires such calculations. Several solution approaches

exist, and these may be grouped into two broad categories. The first is the equation-solving method, and the second is the Gibbs-free-energy-minimization approach.

The equation-solving methods all consider the defining equations of an isothermal flash process as the starting point. The steady-state behavior of an isothermal flash unit may be described by material balances and equilibrium relations. For an n-component system, these equations will form a non-linear algebraic system of 2n equations, some of which may be strongly non-linear. In order to solve such a system, the number of unknown variables must be equal to the number of independent equations. At the outset of most calculations, this condition will not be met unless a sufficient number of the process variables are fixed. Here, we let the specified variables be the feed composition, the temperature and the pressure. Adding a constitutive constraint e.g. that the sum of the mole fractions in each phase must be equal to one, will complete the definition of the system of equations. The remaining 2n+1 variables that need to be calculated are usually the fraction of the feed in the less dense phase (phase split fraction) and the compositions of the two equilibrium phases. Several algorithms have been devised to solve this equation type. They may be conveniently grouped into two broad categories:

equation decoupling methods and global Newton-Raphson methods.

The equation decoupling approach considers a reduced set of equations and unknowns. The defining equations are solved in a specified order, one equation or group of equations at a time, for a corresponding set of variables while holding the remaining variables fixed (Joulia et al., 1985).

One of the earliest decoupling algorithms uses the successive-substitution (SS) method. Null (1970), Prausnitz et al. (1980), Anderson and Prausnitz (1980), Michelson (1982), Nghiem et al. (1983), and Mehra et al. (1983) have discussed its implementation. The number of defining equations is reduced to obtain expressions for the mole fractions in terms of the equilibrium ratio (K-value), feed compositions, and phase split fraction. In effect, the equilibrium ratio and the phase split fraction will replace the two-phase mole fractions as the independent iteration variables. In the inner loop of the algorithm, the phase split fraction is calculated by an iterative technique such as the Newton-Raphson method or the regula-falsi method (Lapidus, 1962). The K-values remain fixed during this process. The outer loop of the algorithm then updates the K-values through any of several techniques. When the new value of the phase split fraction is produced from the inner loop,

it is used to generate new mole fractions and new activity coefficients, where the latter are obtained from an MT model. The equilibrium criteria are then checked, and when they are satisfied to a pre-specified tolerance, convergence is achieved. For strongly non-ideal liquid phases encountered, for example, in liquid-liquid equilibrium problems, the K-values become strong functions of composition, and the convergence of the SS method slows markedly (Michelsen, 1982). Various acceleration techniques have been used in the generation of K-values in the outer loop. Boston and Britt (1978), as part of their "inside-out" algorithm, used Broyden's method (1965) to update the K-values. Prausnitz et al. (1980) used Wegstein's method and Michelsen (1982) used the general-dominant-eigenvalue method of Crowe and Nishio (1975) to accelerate convergence.

The global or multivariable approach simultaneously solves the complete set of descriptive equations in terms of all the process variables. Hirose et al. (1978), Asselineau et al. (1979), Fussel and Yanosik (1978), and Michelsen (1980) discuss the various formulations of equations and choices of independent variables and thermodynamic models that can be combined with a Newton-Raphson iterative scheme. The equations to be solved are grouped into a vector function which is then solved for all the independent

variables. Initial estimates are corrected via a multi-variable Newton-Raphson (NR) method, in which the partial derivatives of all equations with respect to all independent variables are needed. The process is repeated until the equations are simultaneously satisfied. Such evaluations are usually cumbersome and thus time consuming. Jacobians, the matrices that contain the derivative information, are found in one of two ways: perturbation or analytic differentiation (Chimowitz et al., 1982). Often, no simple analytic expressions for the derivatives are available. This is especially true for individual-phase models representing strongly non-ideal systems (Prausnitz et al., 1980). For an n-component system, a finite-difference approach would entail evaluating the vector function at two different values of each of the independent variables. During each of these procedures, the remaining n-1 variables are held constant. The amount of computation is excessive, as the vector function must be evaluated for n+1 sets of independent variables for the n^2 derivatives needed through each iteration. In addition, very good initial estimates are required for this method (Broyden,1965). Although the global approach converges in fewer iterations than the SS method and takes into account K-value compositional dependence, it requires many more computations per iteration.

Quasi-Newton algorithms attempt to improve upon the NR technique by reducing the number of these derivative evaluations. In this class of techniques, partial derivatives are not evaluated directly. Broyden (1965) approximated the Jacobian inverse and then updated this approximation using a secant method. Recognizing that the Jacobian was sparse i.e. contained many zeros, with large systems of non-linear equations, Schubert (1969) found that an approximation to the Jacobian rather than the inverse Jacobian enabled him to take advantage of the zero locations and further reduced derivative evaluations. Lucia and Machietto (1983) combined Newton's method and the Schubert update in their hybrid approach to obtaining derivatives. They divided the Jacobian matrix into a part that was computed analytically and a part that was calculated by the Schubert update. The analytical section contained all the partial derivatives of the total and component mass balances and summation equations, which can be evaluated exactly at little computational expense. The partial derivatives of the ideal-solution parts of the energy and equilibrium equations were also included. The partial derivatives that corresponded to the non-ideal thermodynamic contributions from the energy and equilibrium relations were then placed in the approximated part.

For several examples, between 50% and 70% fewer rigorous physical property calculations were needed in the hybrid algorithm compared to the Newton-Raphson method (Westman et al.,1984). Lucia (1985), Lucia et al. (1985), Joulia et al. (1985), and Venkataraman and Lucia (1986,1987) have studied and modified this hybrid technique.

The Gibbs-free-energy-minimization approach solves the isothermal flash problem in its basic thermodynamic form. At equilibrium, the Gibbs free energy, G, of a system must be at a minimum. G can be expressed as a function of the moles of each component in each phase and their corresponding partial molar free energies. For a two phase equilibrium, this minimization is over the mole numbers in one phase (Grens, 1983). The partial differentials of G with respect to these mole numbers are then equated to zero. The resulting vector of differentials is called the local gradient of the free-energy function. Various gradient minimization methods then solve for the phase compositions that correspond to the minimum of G (Mehra et al, 1980; Heidemann, 1974; Michelsen, 1982). Gautam and Seider (1979) and Ohanomah and Thompson (1984) used the RAND method to solve the problem. After they developed a quadratic approximation to the Gibbs free energy around a current point, an unconstrained objective function was found by

using Lagrange multipliers. A system of linear equations was then solved to find increments for the whole set of variables. Ammar and Renon (1987), in a study of phase split calculations in multicomponent isothermal flash problems, have established that the RAND method has the same convergence behavior as the SS method.

Local Approximations

Leesley and Heyen (1977) interfaced a local approximation with a process simulation package. Their "dynamic straight-line method" or DSL was implemented on the CONCEPT simulation system with the THERMOPACK database and run on a CDC 6600 computer. The function of the DSL was to provide the simulation routines with approximate values of vapor-liquid equilibrium constants. These are obtained by interpolation between carefully selected accurate values provided by the thermodynamic databank.

K-values were represented as simple functions of temperature and pressure with three adjustable parameters per component. As the liquid mixtures considered were relatively ideal (hydrocarbons and light gases), compositional dependence was not considered. These simple forms, which may be envisioned as straight lines plotted in

the region of interest, were 'specialized' by having their parameters fit to data obtained from two iterations of the accurate K-value generator. The modular-sequential simulator then used the LA model through successive iterations until convergence was achieved. In order to assure the accuracy of the solution, a final iteration was carried out using the underlying K-value model. Applications to flowsheet simulations showed that K-values estimated by the LA model seldom departed by more than 5% from accurate data. The following systems were analyzed: (1) an LPG refrigeration plant, (2) a gas treatment plant, and (3) a naphtha separation plant. Reductions of up to 80% in K-value computations and up to 50% in overall simulation computation costs were achieved.

Barrett and Walsh (1979) devised similar models representing vapor and liquid fugacity, activity coefficients, and liquid and gas enthalpies. Their Thermopac Interface or TPIF computer program consisted of a set of subroutines which interfaced between a simulation system and a full thermodynamic computer package and data base (thermopac). TPIF generated physical properties from these simple analytical models which were created and updated with accurate thermodynamic data. More importantly, the equilibrium calculations made use of temperature and

composition derivatives.

A 26-plate acetone/water separator was simulated using the QUASILIN simulation system. QUASILIN characterizes a chemical process by linearizing its equations, solving them by matrix inversion and repeating this scheme iteratively. When TPIF derivatives were not used, the simulation would not converge. The authors found that although an increase of 21% in computer time was seen in the generation of activity coefficients by the simplified model, an overall computer time improvement of 43% occurred. They attribute the total savings to availability of derivatives from simple analytic functions.

The liquid-phase activity-coefficient representation used was based on a regular-solution model with n(n+1) parameters for an n-component mixture. The parameter calculation costs therefore increase quadratically with increase in the number of components. In addition, the underlying MT model used was the relatively simple Wilson equation. Greater savings would probably be found if simpler LA models and more complex MT models were used.

Boston and Britt (1978) have used LA models in quite a different manner. They completely reformulated the multicomponent single-stage problem with their direct-substitution "inside-out" flash algorithm. Accurate phase-

equilibrium and enthalpy models were used to generate parameters for their approximations.

These parameters replaced the conventional iteration variables as the new problem independent variables. The system's describing equations as expressed by these LA models were then solved for the primitive variables by converging a single-variable iteration loop. Control then returned to an outer loop where the model parameters were updated with rigorous equations. Since complex models were only evaluated in an outer iteration loop, less computational effort was required than with conventional procedures. These methods have been applied by Boston (1979) to staged-separator calculations. Potential drawbacks of this technique in process optimization have been discussed by Biegler et al.(1984).

Rebeyrote (1980) developed an LA model procedure tailored to the simulation of a distillation column at steady state. Simple functions of temperature and composition were created to represent K-values of totally miscible mixtures exhibiting ideal-gas behavior. Activity coefficients were approximated by the two-suffix Margules equation for binary mixtures. Subsequently, each multicomponent mixture was treated as a pseudo-binary system formed by the component of interest and a pseudo-component consisting of a combination

of the remaining constituents. The pseudo-binary approach enabled the author to greatly reduce the number of parameters that would have been required to model a 'true' multicomponent mixture e.g. ten parameters would be needed to represent a quartenary system. Rigorous values for liquid-phase activity coefficients were obtained from the NRTL equation for multicomponent systems (Renon and Prausnitz, 1968).

LA models were evaluated for each component on each stage using data points from similar temperature and composition conditions. The required data points were generated from MT evaluations required in previous iterations. The points were chosen according to their temperature and composition distance from the point being approximated. As the iterations neared convergence, the distance critera were 'tightened' and, as a result, a number of points less than the minimum required became available. In this case, the rigorous MT model was used to calculate the equilibrium ratios.

Certain weaknesses in Rebeyrote's LA method contributed to his finding no significant reduction in computational effort. Even though LA models were used for only a third of the calculations, there were still a large number of models and therefore parameters to evaluate. In addition, a

Naphtali and Sandholm (1971) type iterative procedure was implemented where derivatives of the defining equations with respect to the independent variables were required. Derivatives obtained from the simple analytic expressions provided by the LA models were not sufficiently exploited, as temperature derivatives were utilized only in areas where approximations existed, and composition derivatives were not used at all.

By adopting the same approach as Rebeyrote, Stell (1981) was able to address several of these problems. In order to reduce the large number of local models needed by Rebeyrote, Stell utilized fewer LA models for each component in covering the entire distillation column. Only two LA models were usually needed; one for the stripping section and one for the rectifying section. For some columns, only one approximation was needed. For each model, mole fractions and temperatures of data points used to evaluate the parameters were obtained from the previous iteration in the column section of interest. The end points of the column section and data points in between were chosen. The number of points to be used was based on the range of compositions covered; for approximations covering large composition ranges, more data points were used, while for approximations covering narrower composition ranges, fewer points were used. Stell

concluded that use of more than the minimum number of points in parameter evaluations was desirable. He also discovered that, in many cases, diagonal composition derivatives of the equilibrium ratios i.e. the partial derivatives of the activity coefficients of the components of interest with respect to their liquid phase compositions, were crucial to successful convergence.

Stell extended the pseudo-component concept in order to apply his LA approach to azeotropic distillations. To correctly represent mixtures with three widely different component types, a pseudo-ternary two-suffix Margules equation was used. In each particular example, the components were examined and grouped according to chemical type. This grouping problem was not considered in a general way, however, and no attempt was made to automate pseudo-component selection.

The results achieved using LA models were compared with those obtained using the full MT evaluations for equilibrium ratios and derivatives. Reductions of 70%-90% in computational effort were obtained.

Chimowitz et al. (1983, 1984a) detailed the use of LA models in low pressure VLE and LLE calculations. Non-linear approximations for equilibrium ratios for highly nonideal multicomponent systems were developed. As in the works of

Stell and Rebeyrote, the standard state fugacity was approximated by the saturation pressure, and an ideal gas phase for vapor-liquid systems was assumed. The same approach used by Stell and Rebeyrote was also used in accounting for liquid phase non-idealities i.e. a multicomponent mixture was treated as a pseudo-binary mixture using the two-suffix Margules equation. The pseudo-binaries were designated as the pairs formed between a specified reference component (dominant mixture constituent) and each of the other components. The authors used single isothermal flash calculations on a number of non-ideal vapor-liquid systems to demonstrate the superiority of composition-dependent functional approximations over temperature-only models. Since product liquid-phase compositions must lie between the liquid-phase compositions of the bubble and dew points of the feed streams, three or four points spanning this region were used to fit the LA model parameters. For the liquid-liquid systems examined, the parameters were generated from four rigorous liquid-liquid equilibrium calculations i.e. before the computational procedure could be started, the K-value LA models had to be converged over a suitable composition range.

References to the thermodynamic data base were reduced

by at least two orders of magnitude on a number of example systems. Chimowitz et al. (1984b) and Sereno et al. (1986) have applied this procedure to multicomponent distillation columns and liquid-liquid separations, respectively.

Chimowitz and Lee (1984) discuss the LA concept applied to high-pressure processes. Simple analytic functions were created to model ideal systems at moderate pressure. To represent systems at higher pressures, these models were then fit to properties provided by rigorous calculations using an equation of state. A model, representing the equilibrium ratio as a function of temperature and pressure, was compared against the Chao-Seader (1961) correlation in a series of isothermal flash calculations. In a temperture range of 338.9-394.4 K and a pressure range of 480-620 psia, the average deviation between the local and Chao-Seader model was 6.5%. The authors also found substantial time savings in the simulation of a high-pressure distillation column.

Bryan (1985) used LA models to correlate the thermodynamic properties of highly non-ideal liquid mixtures encountered in extraction processes. The pseudo-ternary two-suffix Margules was used to approximate activity coefficients and was applied to single and multi-stage calculations. Generalized procedures were devised for

creating and updating these models.

The foundations of Bryan's LA approach were formulated by considering single-stage calculations. LA models for each of the components in each of the phases of an n-component system were initially created using rigorous thermodynamic properties from the NRTL equation. The three parameters for each of the n-models were evaluated from a number of data points by least-squares regression. Bryan investigated the number and placement of data points to be used for regression, pseudo-component definitions, and model-updating criteria. These will be discussed in more detail in the model development section. These models then completely replaced the underlying thermodynamic equation and provided all activity coefficients and composition derivatives. There are important differences between the LA approach of Chimowitz et al.(1982) and that of Bryan: the creation of the K-value models used by Chimowitz required a number of rigorous flash calculations outside the computational procedure. Bryan used activity coefficient LA models for each component in each phase. These were simply fitted to available properties in the regions of interest, and there was no need for flash calculations. In addition, the LA models of Bryan may be considered to be realistic LA models as they represent small areas in thermodynamic state space.

The K-value expressions of Chimowitz, on the other hand, represent distribution ranges which can hardly be considered local. Bryan's procedure was coupled with a multivariable Newton-Raphson iterative technique and provided derivative information that could have cost as much as n^2 MT evaluations at each iteration. Bryan found that none of the derivatives could be excluded from an efficient scheme in liquid-liquid-equilibrium calculations. This is primarily due to the strong compositional dependence of the K-values. Reductions in the number of MT evaluations of up to 70% were achieved.

The same LA form was also applied to staged-extractor calculations using a Napthali-Sandholm-type iterative technique, but the creation of the LA models was handled differently. Models were first created at the ends of the column. Compositions were then analyzed on subsequent stages and the distance between these and the 'end' composition were calculated. When a distance of more than .15 mole fraction occurred, a new model was created at that location. In this way, two or more models per phase were needed to span the composition region occupied by the extractor. Updating was performed when the same criterion used for single-stage calculations was met. Up to a 95% reduction in rigorous property evaluations was acheived.

Machietto et al. (1986) have recently explored various methods of updating the parameters of LA models. A unique approach to both the mechanics and criteria for updating resulted in improved computational savings. A recursive least-squares procedure required the evaluation of only one rigorous thermodynamic property in updating all the model parameters. Initially, at least as many points as there are parameters are required. In addition, the number of iterations until the next update were adjusted according to an allowed discrepency between the local and rigorous model calculations.

The dynamic simulation of two flash units with a recycle was used as an illustration. With a mixture of acetone, acetonitrile, and water as feed to the process, use of the recursive least-squares algorithm resulted in the reduction of rigorous calculations by a factor of three. In a second example, a wide-boiling ideal mixture of eight hydrocarbons was fed to the same process. A 75% reduction in the number of rigorous bubble-point calculations and an overall computer time reduction of 38% were acheived. In the above two examples, a parameter update was performed at fixed time increments whether the model needed updating or not. Implementing the new 'selective' updating strategy resulted in the prevention of needless modifications to models in,

for example, a steady-state thermodynamic region or in a region adequately covered by the model. The same two examples were run using this 'automatic interval' method. Savings of 80% in rigorous bubble-point calculations and 50% in overall computer time were achieved.

Scope of This Work

This thesis has examined the use of LA models of the Margules type in calculating activity coefficients for non-ideal mixtures. Single-stage phase-equilibrium calculations have been implemented on a number of LLE and VLE systems exhibiting total and partial miscibility. Since non-ideality in the gas phase is not significant at the low pressures considered in this work, all non-idealities will be attributed to the liquid phase. The performance of simple analytic approximations implemented into these calculational procedures have been investigated.

It has been reported (Bryan, 1985) that single-stage calculations are more sensitive to inaccuracies in thermodynamic properties than are multi-stage calculations. In order to avoid the possible 'masking' of results of modifications to the local models, only single-stage calculations have been used. The number and placement of

data points in parameter regression for the various LA models have, therefore, been investigated without concern for the added complexity of a staged process.

The applicability of LA models to distillation and extraction calculations has been considered. Using quantitative results, this study has drawn various conclusions discussing the feasibility of such an approach.

II-MATHEMATICAL DEVELOPMENT

In a single-stage VLE or LLE calculation, the important variables are the temperature, pressure and phase compositions. Once the known and unknown variables have been identified, a suitable scheme is implemented to obtain the quantities of interest.

An important requirement of any calculational strategy is the availability of physical properties. This work considers isothermal flash calculations in which liquid-phase non-idealites are characterized by activity coefficients. The complicated MT model that usually provides these activiy coefficients will be replaced for most of the computation by a simple algebraic equation. This local-approximation (LA) contains a small number of parameters (that are obtained by fitting data-points provided by the MT model). Easily differentiated forms of the LA model will also provide derivatives when they are meeded.

The Single-stage Isothermal Flash Calculation

In an isothermal single-stage equilibrium separation (see figure 1), a feed of molar flow rate F splits into heavy (more dense) and light (less dense) streams with molar

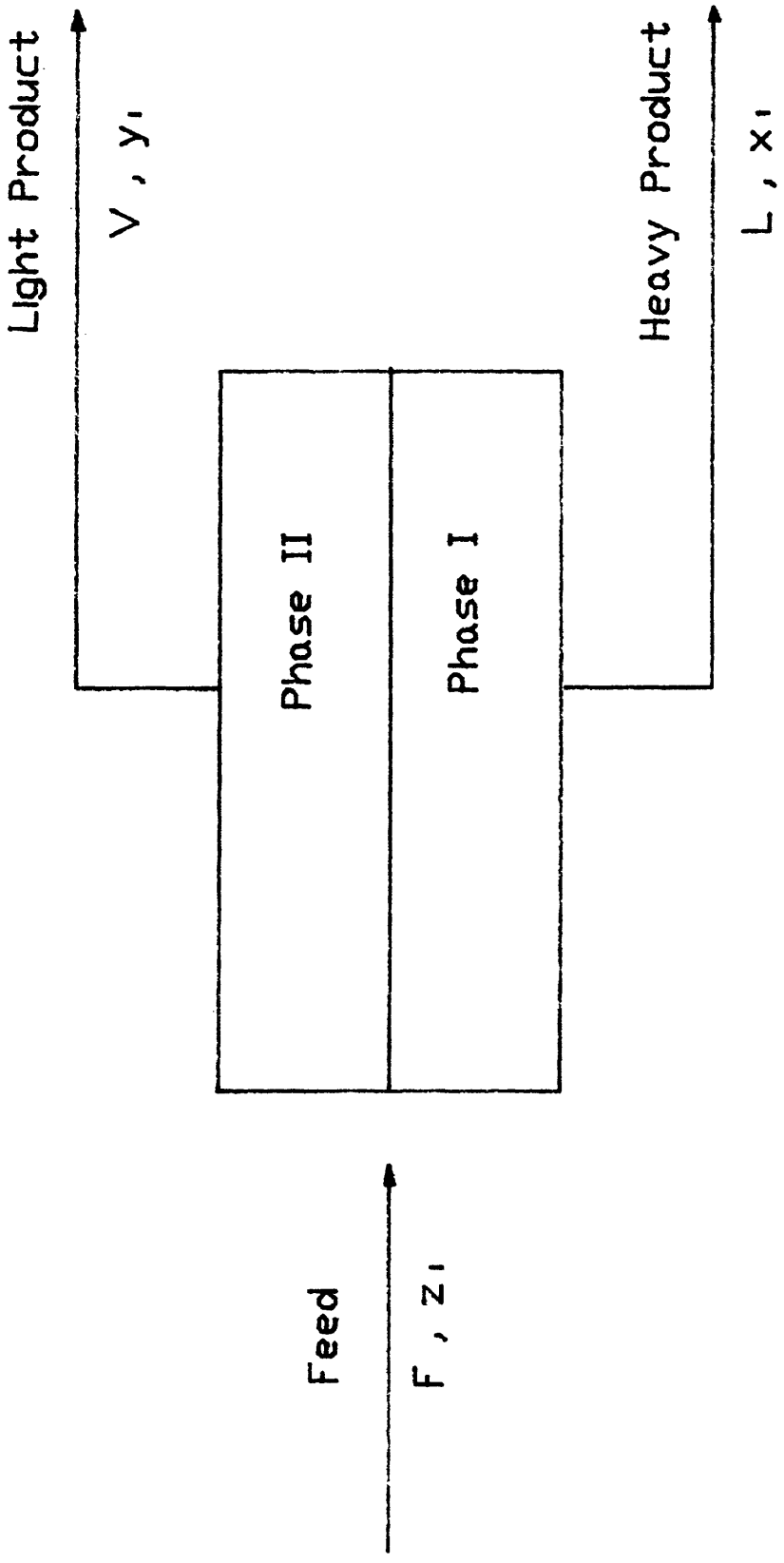


Figure 1. Shematic of A Single Equilibrium Stage

flow rates L and V, respectively. The defining equations for an n-component system include material balances:

$$Fz_i = Vy_i + Lx_i \tag{1a}$$

$$F = V + L \tag{1b}$$

equilibrium restrictions:

$$f_i^V = f_i^L \tag{2}$$

and summation (constitutive) equations:

$$\sum_{i=1}^{n} x_i = 1 \tag{3a}$$

$$\sum_{i=1}^{n} y_i = 1 \tag{3b}$$

where the mole fractions in the feed, light stream, and heavy stream, are, respectively, z_i, y_i, and x_i. Although many combinations and reformulations of equations (3a) and (3b) are possible, only a few have discernible advantages. Therefore, only a few of these rearrangements are used in most computational procedures (Grens, 1983; Ohanomah and

Thompson, 1983). When the flash temperature is specified at the outset of the calculation, the manner in which energy requirements of the process will be satisfied is still unknown. Often, the heat content of the feed is adjusted before it reaches the flash container so that the flash occurs adiabatically (Holland, 1981). In this case, after the isothermal problem is solved, the heat content that the feed must possess is calculated from an enthalpy balance. When the temperature is not specified, an enthalpy balance must be added to the defining equations described above. For a workable solution scheme, the fugacities of the light and heavy phases in equation (2) must first be related to measurable quantities such as temperature, pressure, and composition.

For liquid-liquid equilibrium (LLE) separations, equation (2) becomes:

$$\gamma_i^x x_i = \gamma_i^y y_i \tag{4}$$

where γ_i^y and γ_i^x are the activity coefficients of component i in the light and heavy phases, respectively. For vapor-liquid equilibria (VLE), equation (2) is written:

$$\Phi_i^y y_i P = \gamma_i^x x_i f_i^o \tag{5}$$

where Φ_i^y is the fugacity coefficient of component i in the vapor phase and f_i^o is the standard-state fugacity of component i. If the system pressure P is low or moderately low and the system temperature is below the critical temperatures of all of the components present, all non-idealities may usually be attributed to the liquid phase (Reid et al., 1977) and equation (5) may be simplified to:

$$y_i P = \gamma_i^x x_i P_i^* \tag{6}$$

where P_i^* is the vapor pressure of component i. Equations (1-6) may be rearranged in a number of different ways, though only a few formulations have been used extensively.

Since there are n independent relations for both the material balance and equilibrium equations, and one independent equation of type (3), 2n+1 equations need to be solved simultaneously. When highly non-ideal mixtures are invloved, the equilibrium relations become strongly non-linear functions of composition. Activity coefficients must then be obtained from complex molecular-thermodynamic (MT) models. The computational effort required in a single-stage calculation might not be burdensome, but for a staged separator, the equilibrium must be calculated for each stage

at each iteration. As single-stage computations are often applied to a number of processes e.g. stability analysis (Michelsen, 1981), process flowsheeting (Shacham et al., 1981; Kubicek et al., 1975), simulation of staged processes (Angel et al., 1985), process optimization (Biegler et al., 1984), and dynamic simulation (Hillstad and Hertzberg, 1985; Sereno et al., 1986), devising efficient solution algorithms for them is extremely important.

Solution of Governing Equations by Successive-Substitution

A commonly used arrangement of the equation system is based upon the phase split fraction V/F, which must lie between 0 and 1. If L in equation (1a) is replaced by its definition in equation (1b), and both sides of the equations are divided by F, we obtain:

$$z_i = \alpha y_i + (1-\alpha) x_i \tag{7}$$

where V/F is represented by α. Solving for x_i gives:

$$x_i = \frac{z_i}{(K_i-1)\alpha + 1} \tag{8}$$

where K_i represents γ_i^x/γ_i^y in LLE problems and $(\gamma_i^x P^*)/P$ in

VLE problems. The light phase composition may then be expressed as $y_i=K_i x_i$ or:

$$y_i = \frac{K_i z_i}{(K_i-1)\alpha + 1} \tag{9}$$

One commonly used rearrangement of the constitutive equations is:

$$\sum_{i=1}^{n} (y_i-x_i) = 0 \tag{10}$$

Equations (8) and (9) can now be substituted into equation (10) to give:

$$\sum_{i=1}^{n} \frac{(K_i-1)z_i}{(K_i-1)\alpha +1} = 0 \tag{11}$$

As previously mentioned, it can be shown that only one of the equations in (3) is independent. Equations based on (3a) and (3b) may possess a spurious root and display maxima or minima between α=0 and α=1. These characteristics are often detrimental to iterative techniques. Rachford and Rice (1952) have shown that equation (11) is preferred as it is a

monotonic function of α which always has a negative slope.

The successive-substitution (SS) method is the traditional approach to solving the governing equations (Null, 1970; Prausnitz et al., 1980). In this work, x_i, y_i, and α are chosen as the independent variables and z_i, temperature, and pressure are set at the outset of the calculations. The solution algorithm proceeds as follows:

(1) Initialize x_i and y_i.

(2) Compute K_i^r (x_i^r, y_i^r) from an MT model.

(3) Check $\sum_{i=1}^{n} \left| \gamma_i^x x_i - \gamma_i^y y_i \right| < \varepsilon$? For LLE

$\sum_{i=1}^{n} \left| \gamma_i^x x_i P_i^* - P y_i \right| < \varepsilon$? For VLE

Yes:Converged:Exit.

No: Not Converged: Continue with step (4).

(4) Solve (12) for α with K_i held constant

(5) Set $x_i^{r+1} = z_i / ((K_i^r - 1)\alpha^r + 1)$

(6) Set $y_i^{r+1} = K_i^r x_i^{r+1}$

(7) Return to step (2)

where r represents the iteration number and ε, the convergence tolerance. The inner loop of this algorithm (step (4)) solves for α by iterative techniques such as the Newton-Raphson method (Henley and Rosen, 1969). In the outer loop (steps(2)-(7)), x_i, y_i and α are corrected by an equilibrium ratio (K-value) value obtained in the previous iteration. This method converges rapidly for problems where the K-values are only weakly dependent on the phase compositions. For LLE calculations, this is never the case, with some or all of the K-values being strong functions of both phase compositions. In effect, this "past-composition" method does not account for simultaneous changes in K_i with changes in the other independent variables, and as the sensitivity of K_i to x_i and y_i increases with increasing non-ideality, the number of iterations will increase markedly (Michelsen, 1982; Grens, 1983; Bryan, 1985). Although this computational procedure can be applied to both VLE and LLE systems, we see that it may not always be efficient. By rigorously accounting for the equation

system's multi-dimensional, non-linear nature, the solution of highly non-ideal systems may be better facilitated.

Solution of The Governing Equations by The Multivariable Newton-Raphson Method

The SS approach is described by King (1980) as being a first-order method because the error tends to decrease as the first power of the iteration number. The convergence rate can be improved through the use of a second-order method, where dependence of K_i on phase composition is no longer ignored, and the 2n+1 equations are solved simultaneously for all the independent variables. The multivariable Newton-Raphson (NR) method, which tends to give an error decreasing as the square of the iteration number, is an example of such a second-order technique, and it proceeds as follows:

(1) Set initial values for independent variables and place in row vector $\vec{x}$:

$$\vec{x} = [\ x_1, x_2, \ldots, x_n,\ \ y_1, y_2, \ldots, y_n,\ \ \alpha\ ]$$

(2) Compute the values of all the defining equations

and place in column vector $\vec{G}$:

$$\vec{G} = \begin{bmatrix} z_1-\alpha y_1-(1-\alpha)x_1 \\ z_2-\alpha y_2-(1-\alpha)x_2 \\ \vdots \\ z_n-\alpha y_n-(1-\alpha)x_n \\ f_1^L-f_1^V \\ f_2^L-f_2^V \\ \vdots \\ f_n^L-f_n^V \\ \sum_{i=1}^{n}(y_i-x_i) \end{bmatrix}$$

Check $\sum_{i=1}^{2n+1} \left| \overline{G}_i^r \right| < \varepsilon$

Yes: Converged: Exit

No: Not converged: Proceed to step (3).

(3) Compute $\underline{J}^r$.

(4) Set $\bar{x}^{r+1} = \bar{x}^r - t(\underline{J}^r)^{-1}\bar{G}^r$.

(5) Return to step (2).

The equilibrium criterion equations will assume the form $\gamma_n^x x_n - \gamma_n^y y_n$ for an LLE calculation and $\gamma_n^x x_n P_n^* - P y_n$ for a VLE calculation. This algorithm is described by Lapidus (1962) and Henley and Rosen (1969). $\underline{J}$ is the Jacobian of $\vec{G}$ (see Appendix I) where:

$$\underline{J}_{ij} = \partial \vec{G}_i / \partial \vec{x}_j$$

In order to evaluate $\underline{J}_{ij}$, x_i and y_i must be independently variable i.e. they are not mole fractions and need not sum to one during the iterative calculations.

The scalar quantity t is called the "step-limiting" parameter. Naphtali (1964) has shown that the **direction** of the correction to the variable vector in NR methods (or the relative magnitudes of the elements in vector function $\vec{G}$) is often more reliable than the **magnitude** of the correction. If the **magnitude** of $\vec{G}$ increases from one iteration to the next, then t must assume a value between 0 and 1, in order to

'damp' a possible over-correction. Otherwise, t is set to 1. Kehat and Ghitis (1981) have examined, in great detail, the use of such a parameter for both damping and accelerating the convergence procedure.

This second-order algorithm converges in many fewer iterations than the SS approach. However, due to the requirement that derivatives need to be calculated, fewer convergence iterations do not necessarily result in fewer computations. For each iteration, the SS method requires one evaluation of the activity coefficient for each component in each liquid phase e.g. for a two-liquid-phase system, 2n evaluations are required per iteration. The NR method requires the same number of activity coefficient evaluations as the SS method, but also requires one evaluation of the derivative of each activity coefficient with respect to each component mole number. For two liquid phases, the NR method then requires 2n activity coefficient evaluations and $2n^2$ derivative evaluations per iteration. Thus, in order to choose the superior method for a particular system, the fact that one NR iteration costs about n times as much as an SS iteration must be considered.

Initialization of Independent Variables in LLE Calculations

In order to use the SS or NR convergence methods in LLE phase calculations, initial estimates of the iteration variables will be required. The equilibrium profiles of liquid separations are usually highly nonlinear, and initial estimates that are not very close to the true (converged) phase compositions can lead to divergence and spurious convergence (to the feed composition). The NR method in particular demands an accurate starting point. The convergence requirements of a single variable NR algorithm can be stated as follows (Lapidus, 1962):

(1) If the first and second derivatives of the objective function do not change sign in the interval between the initial value of the iteration variable and its converged value, the iteration will always converge.

(2) If the conditions in (1) do not hold, convergence is not guaranteed.

The multivariable NR algorithm used in this work will have analogous requirements that become more and more restrictive as the dimensionality of the system of equations increases (Bryan, 1985). The efficiency and success of either convergence technique will therefore rely heavily on the initialization used.

Prausnitz and co-workers (1980) noted that the slowly converging variables in isothermal LLE calculations were the mole fractions of the two solvent species in the conjugate liquid phases. These solvent components usually have low mutual solubilities and are present in large amounts. The initialization scheme in this work begins with the determination of the two "solvent species" in the mixture. This method was employed successfully by Prausnitz and co-workers (1980) and proceeds as follows:

(1) For every binary combination i-j in the n component system with both feed compositions z(i) and z(j) greater than .10 :

(a) Create a binary two-phase mixture that can be described by:

$$x_i^I + x_j^I = 1 \qquad (12)$$

$$x_j^J + x_i^J = 1 \qquad (13)$$

where $x_i^I = x_j^J = .98$ and $x_j^I = x_i^J = .02$. I and J denote the two phases.

(b) Assume Phase I is a hypothetical raffinate phase

and calculate the equilibrium ratio of j:

$$K(j) = \frac{\gamma_j^I}{\gamma_j^J} \tag{14}$$

(c) Assume phase J is a hypothetical raffinate phase and calculate the equilibrium ratio of i:

$$K(i) = \frac{\gamma_i^J}{\gamma_i^I} \tag{15}$$

(2) Determine the largest K-value from step(1) and denote the corresponding i or j by E. E denotes "extract solvent" in the mixture.

(3) For every component i where $i \neq E$ and z(i) is greater than .10 :

(a) Create a new binary two-phase mixture:

$$x_i^I + x_E^I = 1 \tag{16}$$

$$x_E^J + x_i^J = 1 \tag{17}$$

where $x_i^I = x_E^J = .98$ and $x_E^I = x_i^J = .02$ and I and J now denote the raffinate and extract phases respectively.

(b) Calculate the K-value of i:

$$K(i) = \frac{\gamma_i^I}{\gamma_i^J} \tag{18}$$

(4) Determine the smallest K-value obtained from step (3) and denote the corresponding i by R. R is the subscript of the "raffinate solvent" in the mixture.

The second part of the initialization process was taken from the work of Bryan (1985). After studying several initialzation schemes for isothermal LLE calculations, he observed that:

(1) Initial estimates of phase compositions located significantly within the heterogeneous region of a mixture caused the NR and SS algorithms to fail by false convergence to the trivial solution (feed composition)

(2) The NR method is tolerant of poor estimates of solute distribution, although failure will occur if the estimates are extremely poor.

Bryan used mutual solubilty estimates to initialize the iteration variables and ensure that the starting values were located outside the two-phase region. After obtaining these solubilities from the solvent selection procedure (see Appendix E), the following steps were taken:

(1) The mole fraction of the "raffinate solvent" in the extract phase is set equal to 75% of its estimated solubility in the "extract solvent", and vice versa.

(2) The solute mole fractions are set equal to their feed compositions or 75% of their estimated solubility in the majority solvent of that phase, whichever is less.

(3) The majority solvent mole fractions in each phase are set so that the mole fractions of that phase sum to one.

(4) Obtain K-values from an activity coefficient correlation and calculate the phase split fraction (α = V/F) by solving the objective function (equation(11)). Obtain values of the component mole fractions from equations (8) and (9).

With this scheme, all ten example systems used in this work converged properly for all flash specifications considered.

Initialization of Independent Variables in VLE Calculations

As VLE systems are often more ideal than LLE systems, a less complicated initialization scheme was obtained from Ohanomah and Thompson (1983). The initial value of the vapor phase fraction (V/F) was obtained by regula-falsi interpolation based on K-values computed from x(i) = y(i) = z(i) (x(i) and y(i) are the liquid and vapor phase compositions and z(i) is the feed composition). For an n component system, an initial interpolation is carried out around f(0) and f(.5) estimated from (Walas, 1982):

$$f(\alpha) = \sum_{i=1}^{n} \frac{(K_i-1)z_i}{1+(k_i-1)\alpha} \tag{19}$$

If the resulting value of α is less than .5, it is accepted as the starting value α_s. If not, a second interpolation is made around f(1.0) and f(.5) estimated from(Walas, 1982):

$$f(\alpha) = \sum_{i=1}^{n} \frac{z_i}{1+(K_i-1)\alpha} - 1 \tag{20}$$

The details of the method are outlined as follows:

(1) Compute K(i) using x(i)=y(i)=z(i)

(2) Compute $F = F_2/(F_2-F_1)$ where:

$$F_1 = \sum_{i=1}^{n} K_i z_i - 1 \tag{21}$$

and:

$$F_2 = 2\sum_{i=1}^{n} \frac{(K_i-1)z_i}{(1+K_i)} \tag{22}$$

(3) If $F > 0$, then $\alpha_s = .5(1-F)$. If not, then $\alpha_s = (F_3-.5F_4)/(F_3-F_4)$ where:

$$F_3 = 2\sum_{i=1}^{n} \frac{z_i}{(1 + K_i)} - 1 \tag{23}$$

and:

$$F_4 = \sum_{i=1}^{n} \frac{z_i}{K_i} - 1 \tag{15}$$

This scheme gave good starting values for all the VLE systems examined.

Formulation of The Local Approximation

A local model must characterize a complex thermodynamic property through simple, functional expressions over limited

regions of composition, temperature, and pressure. Replacing a complex molecular thermodynamic (MT) expression with this local approximation (LA) can result in computational savings. As conditions of composition, temperature, and pressure change, parameters in the local equation may need to be updated. In this way, the approximation will always represent the true thermodynamic behavior of the system as well as possible. As this model must provide reasonably accurate property values with the least amount of computational effort, great care must be taken in its formulation.

The functionality of the LA model must be similar to the MT model. At the low pressures considered in this study, vapor mixtures are considered ideal, and pressure effects on the liquid phase can be neglected. Non-idealities in the liquid phase may be efficiently expressed through the molar excess Gibbs energy, g^E. This quantity is written as a function of the liquid composition at constant temperature and pressure. The activity coefficient γ_i of component i is defined through this function as:

$$RT\ln\gamma_i = \left(\frac{\partial n_t g^E}{\partial n_i}\right)_{T,P,n_{j\neq i}} \tag{25}$$

where R is the universal gas constant, T is the system temperature, P is the system pressure, n_i is the number of moles of component i, and n_t is the total number of moles. The Non-Random Two-Liquid (NRTL) Equation for the excess Gibbs energy (Renon, 1968) is used in this study (see Appendix A). Requiring only binary interaction parameters, it is applicable to multicomponent VLE and LLE systems. A much simpler expession of the excess Gibbs free energy can be obtained if only binary component interactions are considered:

$$g^E = 1/2 \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i^* x_j^* \qquad (26)$$

where the A_{ij} are parameters for two specified components i and j ($A_{ii} = A_{jj} = 0.$ and $A_{ij} = A_{ji}$) and x_i^* and x_j^* are the mole fractions of i and j in the liquid phase. The meaning of x_i^* in this expression is different from its meaning as an independent variable in the iterative calculations. In thermodynamic expressions, the x_i must be normalized:

$$x_i^* = \frac{x_i}{\sum_{j=1}^{n} x_j} \tag{27}$$

where x_i are the composition variables of the iterations, which only need to sum to 1.0 at convergence.

Substituting equation (26) into (25) gives:

$$RT\ln\gamma_i = \sum_{j=1}^{n} \sum_{k=1}^{n} (A_{ij} - (1/2)A_{jk})x_j^* x_k^* \tag{28}$$

For a ternary mixture, equation (28) becomes:

$$RT\ln\gamma_1 = A_{12}(x_2^*)^2 + A_{13}(x_3^*)^2 + (A_{12}+A_{13}-A_{23})x_2^* x_3^* \tag{29}$$

$$RT\ln\gamma_2 = A_{12}(x_1^*)^2 + A_{23}(x_3^*)^2 + (A_{12}+A_{23}-A_{13})x_1^* x_3^* \tag{30}$$

$$RT\ln\gamma_3 = A_{13}(x_1^*)^2 + A_{23}(x_2^*)^2 + (A_{13}+A_{23}-A_{12})x_1^* x_2^* \tag{31}$$

The parameters in each of these equations are calculated by fitting them to data provided by an MT model. Consider equation (29); if more than three sets of x_2^*, x_3^* and γ_1 are obtained from the MT model, equation (29) may be expressed

as a system of three equations in three unknowns (since the combinations of coefficients A_{12}, A_{13} and A_{23} in the last terms of equations (29), (30) and (31) are also constants, they can each be represented by a single term). The parameters may then be solved for by least-squares regression (see Appendix B). The number of parameters in equation (28) increases as n^2, and the determination of each parameter requires at least one data point from the MT model. As a result, for systems of more than three components, use of equation (28) would still involve a great amount of computation.

An alternative is to consider equations (29-31) as "pseudo-ternary" representations (Stell, 1982; Bryan, 1985). Stell and Bryan converted their n-component systems into pseudo-ternaries where the component mole fractions were grouped into one of three pseudo mole fractions.

Consider a generalized form of equations (29-31):

$$RT\ln\gamma_i = A_i(x_j^*)^2 + B_i(x_k^*)^2 + C_i x_j^* x_k^* \qquad (32)$$

where x_i^* is the composition of the component of interest i, and x_j^* and x_k^* are the pseudo compositions into which the n-1

remaining real compositions are placed. The parameters A_i, B_i, and C_i are adjustable and specific to the thermodynamic region in which they were fitted. After a suitable scheme has been devised for the pseudo-component designations, at least three different sets of activity coefficients and phase compositions will be needed for the regression.

The values of x_i, y_i and z_i represent mole fractions only when equations (8), (9) and (11) are satisfied. During the course of a phase-equilibrium calculation, the values taken by x_i, y_i and z_i usually do not sum (for a phase) to unity i.e. they are not mole fractions. These variables may then be regarded as mole numbers approaching mole fractions as convergence is approached (Grens, 1983). Equation (32) can be reformulated as:

$$RT\ln\gamma_i = A_i\left(\frac{x_j^2}{\sigma^2}\right) + B_i\left(\frac{x_k^2}{\sigma^2}\right) + C_i\left(\frac{x_j x_k}{\sigma^2}\right) \qquad (33)$$

where σ represents the summation of all mole numbers:

$$\sigma = \sum_{i=1}^{n} x_i \qquad (34)$$

where x_j and x_k can now vary independently. When the partial derivative of equation (33) is taken with respect to x_i, we obtain:

$$RT\frac{\partial(\ln\gamma_i)}{\partial x_i} = \frac{-2}{\sigma}\left(A_i\frac{x_j^2}{\sigma^2} + B_i\frac{x_k^2}{\sigma^2} + C_i\frac{x_j x_k}{\sigma^2}\right)$$

$$= \frac{-2}{\sigma} RT\ln\gamma_i \qquad (35)$$

Since σ is very near unity near convergence, equation (35) can be rewritten as:

$$\frac{\partial(\ln\gamma_i)}{\partial x_i} \approx -2\ln\gamma_i$$

$$\frac{\partial\gamma_i}{\partial x_i} \approx -2(\gamma_i)\ln\gamma_i \qquad (36)$$

The partial derivative with respect to pseudo-mole number x_j is:

$$RT\frac{\partial(\ln\gamma_i)}{\partial x_j} = \frac{-2}{\sigma}\left(A_i\frac{x_j^2}{\sigma^2} + B_i\frac{x_k^2}{\sigma^2} + C_i\frac{x_j x_k}{\sigma^2}\right)+\frac{1}{\sigma}\left(2A_i\frac{x_j}{\sigma} + C_i\frac{x_k}{\sigma}\right)$$

or:

$$\frac{\partial\gamma i}{\partial x_j} \approx -2(\gamma_i)\ln\gamma_i + \frac{\gamma_i}{RT}\left(2A_i x_j^* + C_i x_k^*\right) \quad (37)$$

The same procedure with pseudo component x_k gives:

$$\frac{\partial\gamma_i}{\partial x_k} \approx -2(\gamma_i)\ln\gamma_i + \frac{\gamma_i}{RT}\left(2B_i x_k^* + C_i x_j^*\right) \quad (38)$$

The diagonal composition derivatives of the $\partial\gamma_i/\partial x_i$ matrix are represented by equation (36) while equations (37) and (38) represent the off-diagonal components. If the number of components in a mixture exceeds three, the off-diagonal derivatives will necessarily appear more than once in the required n x n matrix e.g. if real components #1 and #2 are both given the same pseudo-component designations, the partial derivatives of γ_i with respect to their mole numbers will be the same. As a result, the derivative matrix from a

pseudo-ternary model is always approximate, regardless of the model's mathematical complexity or number of adjustable parameters.

A pseudo-quartenary two-suffix Margules equation can be derived from equation (28) and written as:

$$RT\ln\gamma_i = A_i(x_j^*)^2 + B_i(x_k^*)^2 + C_i(x_m^*)^2 + D_i x_j^* x_k^* + E_i x_j^* x_m^* + F_i x_k^* x_m^* \qquad (39)$$

where,in addition to the terms in equation (32), pseudo-composition x_m^* is now included. Equation (39) is expected to provide more accurate estimates of activity coefficients and their composition derivatives, but at the expense of increased computational cost; regression for six parameters will require activity coefficients for at least six different sets of phase compositions. The composition derivative expressions are obtained from equation (39) in the same manner as they were obtained from equation (32) (see appendix G).

Pseudo-Composition Definitions in The LA Model

If a pseudo-binary Margules expression is used, there is no ambiguity in grouping the real components into pseudo-components. All components other than the component of

interest are simply grouped into a single pseudo-composition. With the pseudo-ternary and pseudo-quartenary models, there are either two or three pseudo-compositions into which the real components can be placed. An efficient grouping strategy must therefore be devised.

Bryan (1985) used K-values as a grouping criterion for the following reasons:

1) K-values are obtained during the calculational procedure. A general grouping scheme built into the procedure is possible when these K-values are used.

2) When the NR iteration technique is used, n x n composition derivatives of the activity coefficients are required (for an n-component system). Bryan noted that when there was a large difference in magnitude or sign among the off-diagonal elements in a row of the $\partial\gamma_i/\partial x_j$ matrix, derivatives with respect to the components for which $K_j > 1.0$ usually had similar values. Derivatives with respect to the components for which $K < 1.0$ also had similar values.

The pseudo-ternary and pseudo-quartenary two-suffix Margules equations will provide only three and four unique values for the composition derivatives of a component activity coefficient, respectively. The task of grouping the components into pseudo-components is a crucial one, as the best approximation to the derivatives will produce the best

estimates of the activiy coefficients.

Using current estimates of K-values, the following grouping strategy was implemented with the pseudo-ternary LA model:

1) Consider an n-component mixture. Let component i be the component for which we wish to create a model i.e. fit parameters.
2) All components other than i whose K-values are greater than 1.0 are found. Their compositions are then grouped into a pseudo-component composition x_B.
3) All components other than i whose K-values are less than 1.0 are found. Their compositions are then combined into a pseudo-component composition x_C.
4) If the component of interest is one of the solvent species IL or IV, it will sometimes happen that either x_B or x_C will be set to zero. The pseudo-ternary will then collapse into a pseudo-binary . Bryan (1985) found that this often degrades the performance of the LA model when non-ideal LLE mixtures are considered. The following procedure is then implemented.
    a) If the component of interest,i, is the

minority solvent (IL or IV) of the phase of interest, then the composition of the component (other than i) whose K-value is the least is grouped into x_C.

b) If the component of interest,i, is the majority solvent (IL or IV) of the phase of interest, then the composition of the component (other than i) whose K-value is the greatest is grouped into x_B.

Using current estimates of K-values, the following grouping scheme was implemented with the pseudo-quartenary model:

1) Let component i be the component of interest.
2) If a component (other than i) is the 'extract' solvent, IV, or if the K-value of this component is greater then the K-value of IV, then its real composition is grouped into pseudo-component x_B.
3) If a component (other than i) is the 'raffinate' solvent, IL, or has a K-value less than the K-value of IL, then its composition is grouped into pseudo-component x_C.
4) All remaining real compositions (other than i) are grouped into pseudo-composition x_D.

5) When the component of interest, i, is one of the solvent species(IV or IL) it will sometimes happen that x_B, x_C or x_D will be set equal to zero. The quartenary form will then be reduced to a ternary form. As this ternary model performs well for all the mixtures considered, this will be allowed to occur.

For an n-component system that distributes into two phases, 2n sets of pseudo-composition definitions will be created for fitting parameters to 2n different local approximations.

These rules will also apply to VLE systems where the solvent species IV and IL are now identified as the key components in the mixture i.e. the component with the highest K-value will be IV and the component with the lowest K-value will be IL.

Taylor Series Expansion and Moving Base Point Strategy

When an LA model is being used, the iteration will converge to a solution consistent with that particular model. The true solution,i.e. the one obtained with activity coefficients from the complex MT model, will not be exactly equal to the 'LA solution'. Some further iteration will

therefore be required. One approach is to use the MT correlation somehow. This would entail implementing the NR or SS algorithm with full MT thermodynamics, or using the two iteration techniques with partial MT thermodynamics. These schemes are considered in more detail in a later chapter.

Another approach to arriving at the 'true solution' uses Taylor series expansion forms of the pseudo-ternary and quartenary two-suffix Margules equations. Appendix D details the derivation of these equations, and Appendix H contains their composition derivatives. These may be respectively written as:

$$\begin{aligned} RT\ln\gamma_i = {} & RT\ln\gamma_i^\circ + C1(x_B^*-(x_B^*)^\circ) + C2(x_C^*-(x_C^*)^\circ) \\ & + C3(x_B^*-(x_B^*)^\circ)^2 + C4(x_C^*-(x_C^*)^\circ)^2 \\ & + C5(x_B^*-(x_B^*)^\circ)(x_C^*-(x_C^*)^\circ) \end{aligned} \tag{40}$$

and:

$$\begin{aligned} RT\ln\gamma_i = {} & RT\ln\gamma_i^\circ + C1(x_B^*-(x_B^*)^\circ) + C2(x_C^*-(x_C^*)^\circ) \\ & + C3(x_D^*-(x_D^*)^\circ) + C4(x_B^*-(x_B^*)^\circ)^2 \\ & + C5(x_C^*-(x_C^*)^\circ)^2 + C6(x_D^*-(x_D^*)^\circ)^2 \\ & + C7(x_B^*-(x_B^*)^\circ)(x_C^*-(x_C^*)^\circ) \\ & + C8(x_B^*-(x_B^*)^\circ)(x_D^*-(x_D^*)^\circ) \\ & + C9(x_C^*-(x_C^*)^\circ)(x_D^*-(x_D^*)^\circ) \end{aligned} \tag{41}$$

where x_B^*, x_C^* and x_D^* are the pseudo-component mole fractions and $(x_B^*)^\circ$, $(x_C^*)^\circ$, and $(x_D^*)^\circ$ are the pseudo-component mole fractions around which the original expression was expanded (base point). γ_i° represents the true activity coefficients evaluated at this base point. The potential superiority of equations (40) and (41) over equations (32) and (39) is that they may be manipulated to arrive at the 'true' MT solution without using MT properties (with the exception of MT activity coefficients needed for regression analysis).

The moving base point strategy takes advantage of the the fact that equations (40) and (41) are 'perfect' at the base point e.g. when $x_B^* = (x_B^*)^\circ$ and $x_C^* = (x_C^*)^\circ$ in equation (40), then $\gamma_i = \gamma_i^\circ$. The initial solution to which an iteration scheme utilizing an expansion form will converge might not be extremely close to the true solution. However, if the base point now becomes this 'first-solution' and the model parameters are refitted, the equation once again becomes perfect at this point. If this process is repeated only a small number of times, the 'true-solution' should be obtained with computational savings.

III - DISCUSSION OF RESULTS

In order to test the effectiveness of the various LA methods, a number of flash calculations were performed. Given feed compositions, these calculations will solve for isobaric, isothermal equilibrium phase compositions. Five test cases for each type of flash (VLE or LLE) were used and are listed in table 1. The systems represent a variety of types of mixtures used in extraction and distillation processes.

Description of VLE Systems

The vapor phase is assumed ideal for the VLE mixtures considered , and the liquid phase is modeled by an activity coefficient correlation. This assumption prohibits inclusion of components that tend to dimerize or polymerize in the vapor phase such as organic acids. The components in the VLE mixtures have been chosen to display marked liquid phase non-ideality.

System 1, an important system in Fischer-Tropsch processes(Griswold and Buford,1949) exhibits strongly non-ideal behavior. The carbonyl groups on the acetone molecules will form hydrogen bonds with methanol and water. Methanol

Table 1. Multicomponent Systems Investigated

System #	Type	Components(indices)	T(K)
1	VLE	Acetone(1)-Methanol(2) -Water(3)	338.15
2	VLE	Hexane(1)-Benzene(2) -Ethanol(3)	333.15
3	VLE	Methanol(1)-Ethanol(2) -Acetone(3)-Water(4)	338.15
4	VLE	Cyclohexane(1)-Benzene(2) -Toluene(3)-Ethanol(4)	343.15
5	VLE	Hexane(1)-Heptane(2) -Cyclohexane(3)-Benzene(4) -Toluene(5)-Ethanol(6)	343.15
6	LLE	Acetonitrile(1)-Methanol(2) -n-octane(3)	298.15
7	LLE	N-N-Dimethylforamide(DMF)(1) -Water(2)-Benzene(3)	293.15
8	LLE	Heptane(1)-DMF(2) -Water(3)-Benzene(4)	293.15
9	LLE	Iso-octane(1)-Cyclohexane(2) -Benzene(3)-Furfural(4)	293.15
10	LLE	Benzene(1)-Toluene(2) -Cyclohexane(3)-Hexane(4) -Heptane(5)-Sulfolane(6)	348.15

and water will also associate (form bonds with molecules of their own type). System 2 is a solution of an alcohol in a non-polar fluid. The effects of alcohols on the separation of n-hexane-benzene mixtures has been discussed in the literature(Govindaswamy et al.,1977). When the alcohol is present in dilute quantities, nearly all its molecules are monomers. When its concentration increases, the aformentioned phenomenon of association causes polymerization to occur. Systems 3, 4 and 5 are essentially combinations of the previous mixtures.

Description of LLE Systems

The LLE ternary systems have been chosen to represent 'type I' and 'type II' mixtures (Ternary mixtures with only one partially miscible binary are called type I systems, and type II systems are designated as those with two partially miscible binaries having only one region of immiscibility). System 6 represents a type II mixture in which acetonitrile -methanol and methanol-n-octane are the partially miscible pairs. The components of System 6 represent three classes of organic compounds with different interaction energies i.e. paraffins displaying dispersional interactions, acetonitrile showing polar interactions, and alcohols which show specific

interaction forces (hydrogen-bonding) (Kikic,I. et al., 1982). System 7 exhibits type I behavior where benzene and water are the partially miscible pair.

System 10 is a typical six-component aromatics extraction system (Mukhopadhyay and Sahasranaman, 1982) where sulfolane is used as the selective polar solvent. Sulfolane is used in the important commercial process of separating aromatic hydrocarbons from multicomponent hydrocarbon feedstocks such as reformed naphtha. System 10 along with Systems 8 and 9 can be reduced to ternary systems that exhibit type I and type II behaviour. These, along with Systems 6 and 7 are graphically represented in figures 2 through 5.

Convergence of Single-Stage LLE and VLE Calculations by Successive Substitution

Single-stage isothermal flash calculations were performed on the ten test systems presented in table 1. NRTL parameters were needed for every binary in each of these systems(see Appendix C). After the iteration variables are initialized by the methods described in chapter II, they are converged by the successive-substitution approach to a tolerance of 5×10^{-5}. This convergence tolerance is the sum

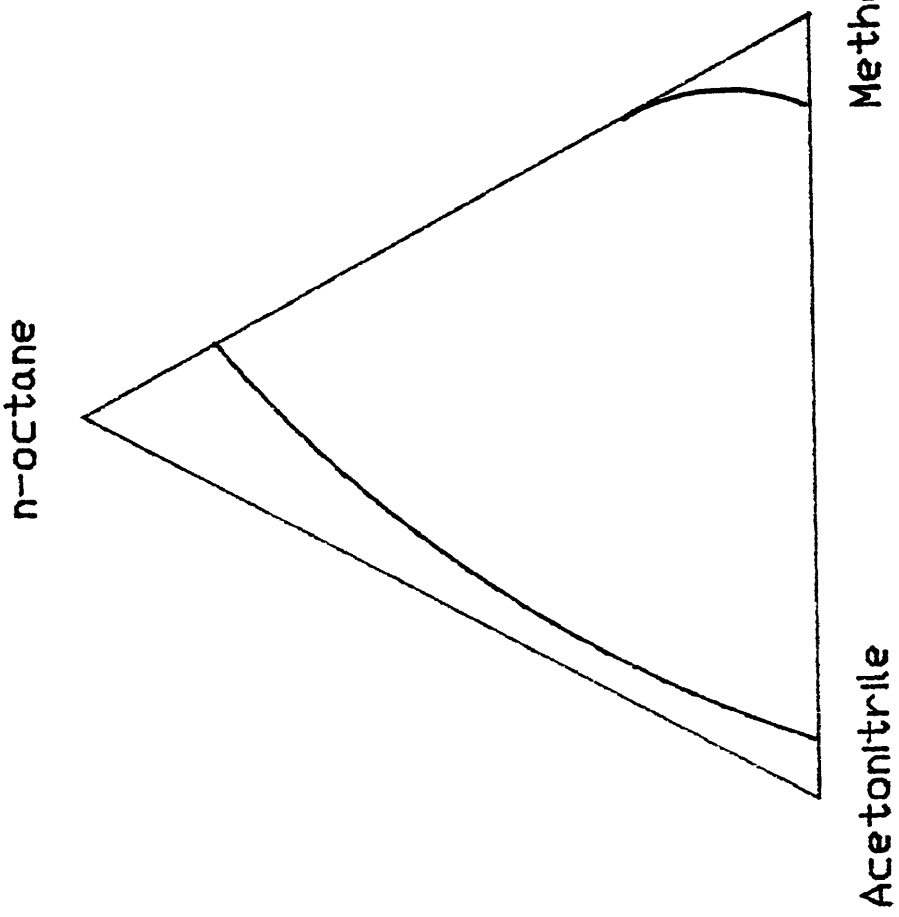


Figure 2. Phase Diagram for Ternary System 6

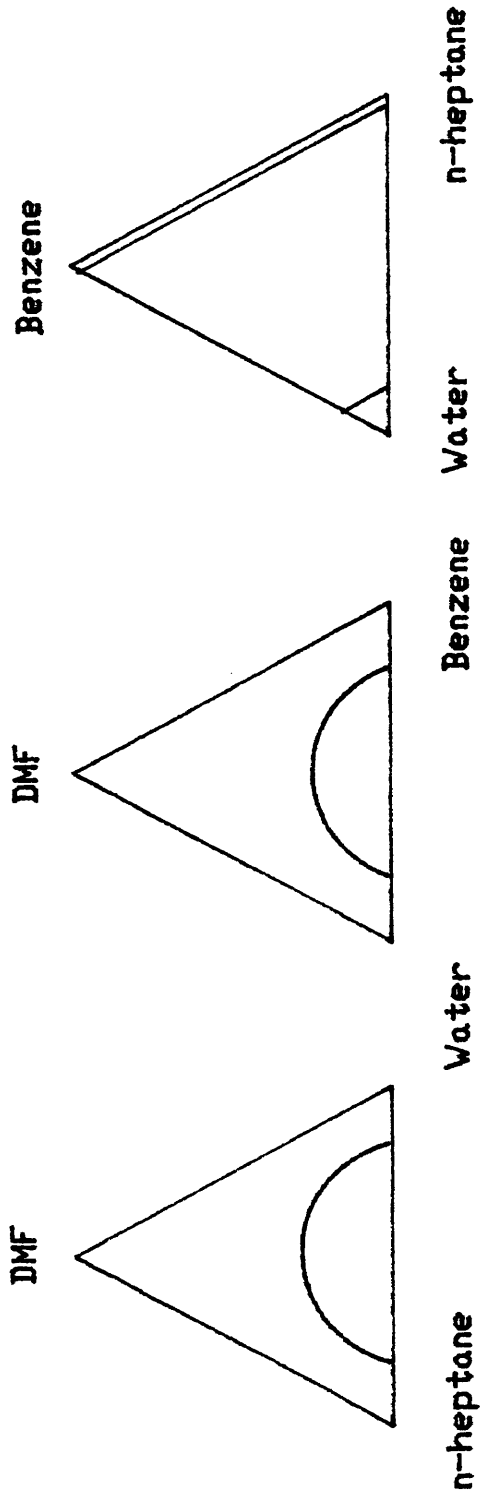


Figure 3. Phase Diagrams for The Partially Miscible Ternary Sub-systems of System 8

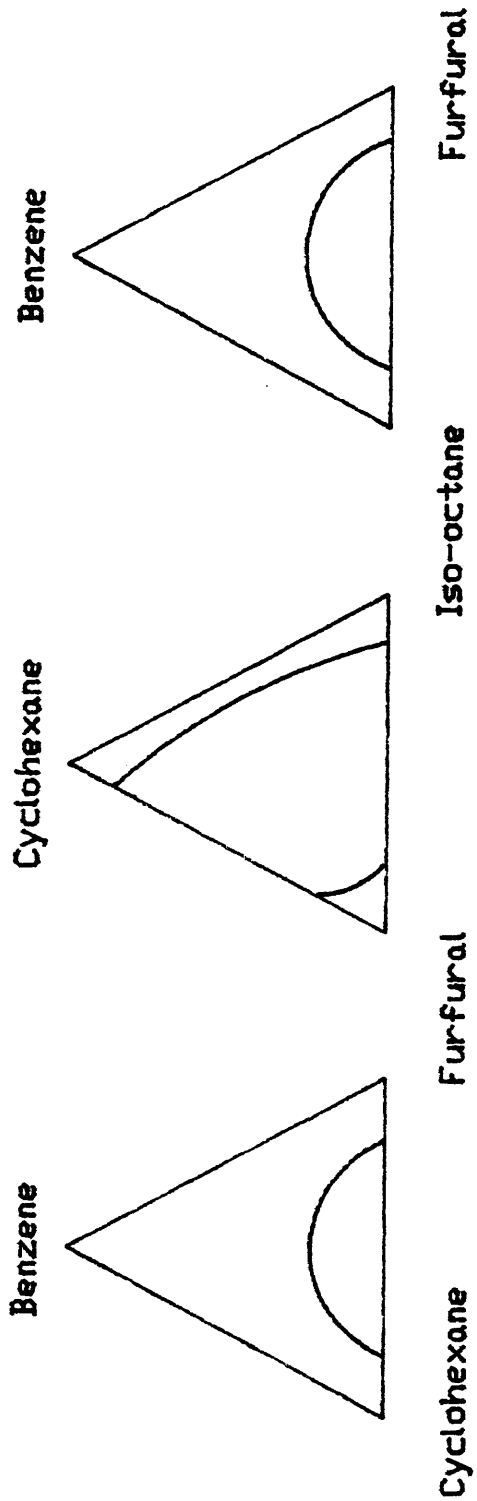


Figure 4. Phase Diagrams of Partially Miscible Ternary Sub-systems of System 9

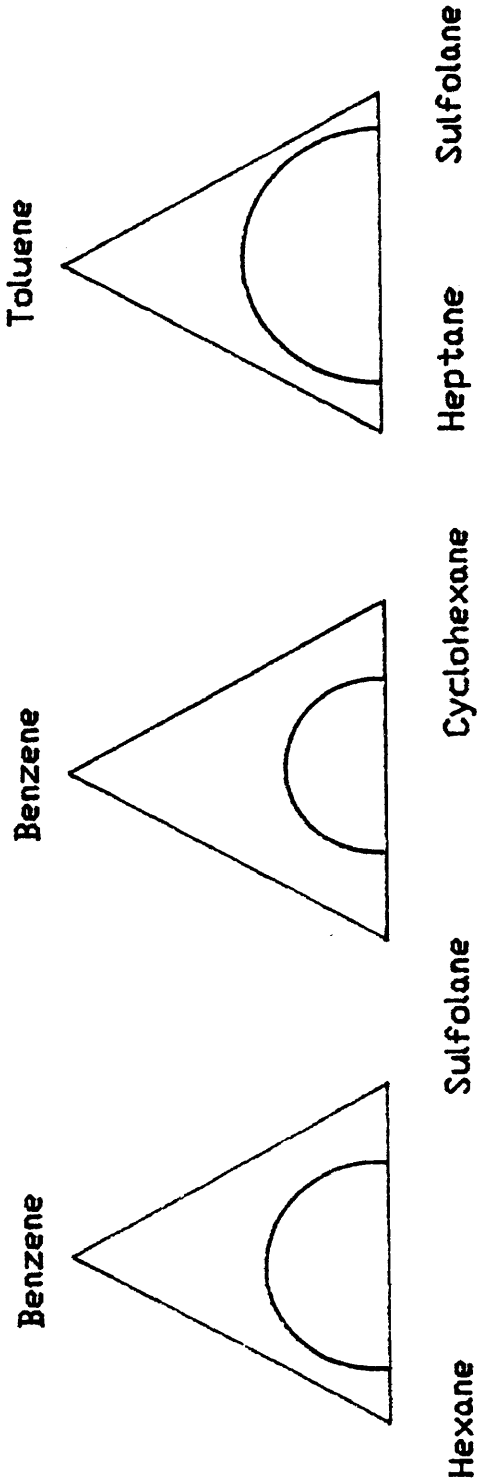


Figure 5. Phase Diagrams for Partially Miscible Ternary Sub-systems of System 10

of the absolute values of the (2n+1) elements of the objective function. The results of fifty such calculations are presented in tables 2 - 11.

The SS method is very reliable but is often very 'expensive'. For an n-component system, each iteration requires n (VLE) or 2n (LLE) values of the activity coefficients. When these properties are obtained from an MT model like the NRTL equation and the number of SS iterations is high, the computational load becomes burdensome. Convergence rates tend to be highest when the SS iteration is far from the solution, and as the solution is approached , 'creeping' often occurs (Bryan, 1985). This aspect often makes the SS approach ineffective. Grens (1983) and Heidemann(1983) discuss this shortcoming by introducing the 'spectral radius of the convergence matrix' which governs SS convergence behaviour.

Convergence of Single-Stage LLE and VLE Calculations by The Multivariable Newton-Raphson Method

The multivariable NR method displays second-order convergence. It will almost always converge to a solution in less iterations than the SS method. However, the number of property evaluations through each iteration is now

Table 2. Results of VLE Calculations by Successive-Substitution for Ternary System Acetone-Methanol-Water

System 1: Acetone(1) Methanol(2) Water(3)
(T = 338.15 K)

Ex. #	Cmpt. #	z_i	x_i	y_i	K_i	α	SS Iterations to Converge
1	1	0.350	0.267	0.588	2.200	0.2581	17
	2	0.250	0.249	0.252	1.010		
	3	0.400	0.484	0.159	0.329		
2	1	0.400	0.277	0.627	2.270	0.3518	22
	2	0.200	0.198	0.204	1.030		
	3	0.400	0.525	0.169	0.321		
3	1	0.450	0.282	0.667	2.370	0.4371	28
	2	0.150	0.146	0.155	1.060		
	3	0.400	0.572	0.178	0.312		
4	1	0.500	0.277	0.708	2.560	0.5174	38
	2	0.100	0.095	0.104	1.100		
	3	0.400	0.628	0.188	0.299		
5	1	0.500	0.254	0.750	2.950	0.4959	53
	2	0.050	0.046	0.054	1.170		
	3	0.450	0.700	0.196	0.280		

Table 3. Results of VLE Calculations by Successive-Substitution for Ternary System Hexane-Benzene-Ethanol

System 2: Hexane(1) Benzene(2) Ethanol(3)
(T = 333.15 K)

Ex. #	Cmpt. #	z_i	x_i	y_i	K_i	α	SS Iterations to Converge
1	1	0.700	0.799	0.628	0.786	0.5763	17
	2	0.100	0.122	0.084	0.692		
	3	0.200	0.080	0.288	3.610		
2	1	0.650	0.723	0.582	0.805	0.5178	21
	2	0.150	0.181	0.121	0.672		
	3	0.200	0.096	0.296	3.080		
3	1	0.600	0.644	0.539	0.838	0.4192	23
	2	0.200	0.234	0.153	0.657		
	3	0.200	0.123	0.307	2.500		
4	1	0.550	0.601	0.519	0.864	0.6272	29
	2	0.200	0.255	0.167	0.654		
	3	0.250	0.143	0.313	2.190		
5	1	0.500	0.526	0.494	0.938	0.8092	42
	2	0.200	0.275	0.182	0.664		
	3	0.300	0.199	0.324	1.630		

Table 4. Results of VLE Calculations by Successive-Substitution for Quaternary System Methanol-Ethanol-Acetone-Water

System 3: Methanol(1) Ethanol(2) Acetone(3) Water(4)
(T = 338.15 K)

Ex. #	Cmpt. #	z_i	x_i	y_i	K_i	α	SS Iterations to Converge
1	1	0.112	0.111	0.114	1.020	0.4664	16
	2	0.112	0.128	0.095	0.745		
	3	0.425	0.267	0.606	2.270		
	4	0.350	0.494	0.185	0.375		
2	1	0.087	0.085	0.090	1.060	0.5773	22
	2	0.087	0.098	0.080	0.811		
	3	0.475	0.258	0.634	2.460		
	4	0.350	0.560	0.197	0.351		
3	1	0.013	0.009	0.013	1.520	0.7939	33
	2	0.013	0.008	0.014	1.600		
	3	0.625	0.147	0.749	5.090		
	4	0.350	0.836	0.224	0.268		
4	1	0.038	0.032	0.042	1.280	0.5573	37
	2	0.038	0.034	0.040	1.190		
	3	0.475	0.189	0.702	3.720		
	4	0.450	0.745	0.216	0.290		
5	1	0.200	0.202	0.199	0.984	0.5545	9
	2	0.200	0.252	0.159	0.631		
	3	0.400	0.259	0.514	1.990		
	4	0.200	0.288	0.129	0.448		

Table 5. Results of VLE Calculations by Successive-Substitution for Quaternary System Cyclohexane-Benzene-Toluene-Ethanol

System 4: Cyclohexane(1) Benzene(2) Toluene(3) Ethanol(4)
(T = 343.15 K)

Ex. #	Cmpt. #	z_i	x_i	y_i	K_i	α	SS Iterations to Converge
1	1	0.230	0.230	0.233	1.020	0.1230	14
	2	0.320	0.328	0.266	0.812		
	3	0.210	0.230	0.068	0.294		
	4	0.240	0.213	0.433	2.030		
2	1	0.300	0.322	0.289	0.899	0.6721	17
	2	0.300	0.352	0.274	0.779		
	3	0.100	0.195	0.054	0.276		
	4	0.300	0.131	0.382	2.920		
3	1	0.200	0.189	0.203	1.080	0.7870	24
	2	0.300	0.346	0.288	0.832		
	3	0.100	0.222	0.067	0.301		
	4	0.400	0.243	0.442	1.820		
4	1	0.400	0.455	0.394	0.866	0.8956	26
	2	0.100	0.125	0.097	0.777		
	3	0.100	0.269	0.080	0.298		
	4	0.400	0.151	0.429	2.840		
5	1	0.300	0.312	0.298	0.954	0.8592	26
	2	0.200	0.242	0.193	0.798		
	3	0.100	0.254	0.075	0.294		
	4	0.400	0.191	0.434	2.270		

Table 6. Results of VLE Calculations by Successive-Substitution for Six Component System Hexane-Heptane-Cyclohexane-Benzene-Toluene-Ethanol

System 5: Hexane(1) Heptane(2) Cyclohexane(3) Benzene(4) Toluene(5) Ethanol(6)
(T = 343.15 K)

Ex. #	Cmpt. #	z_i	x_i	y_i	K_i	α	SS Iterations to Converge
1	1	0.150	0.144	0.173	1.200	0.1926	15
	2	0.175	0.196	0.086	0.441		
	3	0.167	0.173	0.142	0.823		
	4	0.167	0.173	0.138	0.797		
	5	0.175	0.202	0.060	0.298		
	6	0.167	0.111	0.400	3.600		
2	1	0.100	0.094	0.115	1.230	0.2817	16
	2	0.150	0.177	0.080	0.453		
	3	0.200	0.210	0.175	0.834		
	4	0.200	0.212	0.169	0.799		
	5	0.150	0.187	0.055	0.292		
	6	0.200	0.119	0.405	3.390		
3	1	0.075	0.069	0.086	1.240	0.3276	17
	2	0.138	0.167	0.077	0.460		
	3	0.217	0.229	0.192	0.839		
	4	0.217	0.232	0.185	0.800		
	5	0.138	0.179	0.052	0.289		
	6	0.217	0.124	0.408	3.300		
4	1	0.050	0.046	0.057	1.260	0.3747	18
	2	0.125	0.156	0.073	0.468		
	3	0.233	0.248	0.209	0.845		
	4	0.233	0.252	0.202	0.801		
	5	0.125	0.171	0.049	0.286		
	6	0.233	0.128	0.410	3.210		
5	1	0.025	0.022	0.029	1.280	0.4237	19
	2	0.112	0.145	0.069	0.476		
	3	0.250	0.267	0.227	0.851		
	4	0.250	0.273	0.219	0.801		
	5	0.112	0.162	0.046	0.283		
	6	0.250	0.132	0.411	3.120		

Table 7. Results of LLE Calculations by Successive-Substitution for Ternary System Acetonitrile-Methanol-n-octane

System 6: Acetonitrile(1) Methanol(2) n-octane(3)
(T = 298.15 K)

Ex. #	Cmpt. #	z_i	x_i	y_i	K_i	α	SS Iterations to Converge
1	1	0.175	0.290	0.023	0.079	0.4303	14
	2	0.450	0.640	0.198	0.309		
	3	0.375	0.070	0.779	1.120		
2	1	0.150	0.237	0.021	0.082	0.4037	16
	2	0.500	0.682	0.231	0.338		
	3	0.350	0.080	0.749	9.340		
3	1	0.125	0.188	0.019	0.101	0.3732	19
	2	0.550	0.719	0.266	0.370		
	3	0.325	0.093	0.715	7.690		
4	1	0.100	0.142	0.017	0.119	0.3359	23
	2	0.600	0.749	0.306	0.409		
	3	0.300	0.109	0.677	6.190		
5	1	0.075	0.099	0.015	0.146	0.2862	31
	2	0.650	0.769	0.352	0.458		
	3	0.275	0.131	0.633	4.820		

Table 8. Results of LLE Calculations by Successive-Substitution for Ternary System N,N-Dimethylformamide-Water-Benzene

System 7: DMF(1) Water(2) Benzene(3)
(T = 293.15 K)

Ex. #	Cmpt. #	z_i	x_i	y_i	K_i	α	SS Iterations to Converge
1	1	0.175	0.211	0.107	0.506	0.3493	17
	2	0.512	0.775	0.023	0.030		
	3	0.313	0.013	0.870	64.90		
2	1	0.200	0.237	0.129	0.547	0.3429	23
	2	0.500	0.745	0.030	0.041		
	3	0.300	0.018	0.840	46.30		
3	1	0.225	0.261	0.154	0.591	0.3370	31
	2	0.488	0.715	0.040	0.056		
	3	0.287	0.024	0.806	33.50		
4	1	0.250	0.284	0.182	0.640	0.3319	45
	2	0.475	0.685	0.053	0.078		
	3	0.275	0.032	0.765	24.30		
5	1	0.260	0.293	0.194	0.662	0.3376	55
	2	0.465	0.671	0.060	0.089		
	3	0.262	0.035	0.745	21.20		

Table 9. Results of LLE Calculations by Successive-Substitution for Quartenary System Heptane-N,N-Dimethylformamide-Water-Benzene

System 8: Heptane(1) DMF(2) Water(3) Benzene(4)
(T = 293.15 K)

Ex. #	Cmpt. #	z_i	x_i	y_i	K_i	α	SS Iterations to Converge
1	1	0.220	0.004	0.441	10.80	0.4940	18
	2	0.260	0.423	0.093	0.210		
	3	0.260	0.509	0.005	0.009		
	4	0.260	0.064	0.461	7.170		
2	1	0.230	0.010	0.428	43.80	0.5264	29
	2	0.290	0.479	0.120	0.250		
	3	0.190	0.394	0.006	0.002		
	4	0.290	0.116	0.446	3.830		
3	1	0.200	0.010	0.387	40.40	0.5039	32
	2	0.300	0.472	0.131	0.277		
	3	0.200	0.395	0.008	0.019		
	4	0.300	0.123	0.474	3.860		
4	1	0.175	0.016	0.344	22.10	0.4862	46
	2	0.325	0.483	0.158	0.327		
	3	0.175	0.330	0.011	0.033		
	4	0.325	0.171	0.486	2.850		
5	1	0.150	0.026	0.301	11.50	0.4501	68
	2	0.350	0.482	0.189	0.392		
	3	0.150	0.260	0.015	0.059		
	4	0.350	0.232	0.494	2.130		

Table 10. Results of LLE Calculations by Successive-Substitution for Quartenary System Iso-octane-Cyclohexane-Benzene-Furfural

System 9: Iso-octane(1) Cyclohexane(2) Benzene(3) Furfural(4)
(T = 293.15 K)

Ex. #	Cmpt. #	z_i	x_i	y_i	K_i	α	SS Iterations to Converge
1	1	0.325	0.056	0.465	8.260	0.6581	24
	2	0.175	0.051	0.239	4.660		
	3	0.175	0.150	0.188	1.250		
	4	0.325	0.742	0.108	0.146		
2	1	0.300	0.059	0.406	6.900	0.6945	28
	2	0.200	0.063	0.260	4.100		
	3	0.200	0.173	0.212	1.230		
	4	0.300	0.705	0.122	0.173		
3	1	0.275	0.062	0.350	5.660	0.7404	33
	2	0.225	0.078	0.277	3.560		
	3	0.225	0.196	0.235	1.200		
	4	0.275	0.664	0.138	0.208		
4	1	0.250	0.065	0.295	4.520	0.8025	41
	2	0.250	0.095	0.288	3.030		
	3	0.250	0.220	0.257	1.170		
	4	0.250	0.620	0.159	0.256		
5	1	0.125	0.075	0.301	4.030	0.2222	49
	2	0.125	0.090	0.248	2.770		
	3	0.250	0.242	0.279	1.160		
	4	0.500	0.594	0.171	0.288		

Table 11. Results of LLE Calculations by Successive-Substitution for Six Component System Benzene-Toluene-Cyclohexane-Hexane-Heptane-Sulfolane

System 10: Benzene(1) Toluene(2) Cyclohexane(3) Hexane(4) Heptane(5) Sulfolane(6)
(T = 348.15 K)

Ex. #	Cmpt. #	z_i	x_i	y_i	K_i	α	SS Iterations to Converge
1	1	0.100	0.116	0.072	0.626	0.3633	11
	2	0.100	0.117	0.071	0.604		
	3	0.200	0.296	0.032	0.108		
	4	0.200	0.304	0.018	0.060		
	5	0.100	0.154	0.006	0.038		
	6	0.300	0.014	0.801	56.50		
2	1	0.200	0.219	0.148	0.674	0.2713	17
	2	0.200	0.222	0.140	0.631		
	3	0.133	0.172	0.030	0.173		
	4	0.133	0.176	0.018	0.104		
	5	0.133	0.178	0.013	0.071		
	6	0.200	0.032	0.651	20.30		
3	1	0.300	0.327	0.236	0.722	0.2988	26
	2	0.200	0.221	0.151	0.683		
	3	0.100	0.129	0.031	0.242		
	4	0.100	0.134	0.021	0.159		
	5	0.100	0.136	0.016	0.118		
	6	0.200	0.053	0.544	10.20		
4	1	0.275	0.286	0.210	0.735	0.1425	32
	2	0.275	0.287	0.200	0.697		
	3	0.108	0.121	0.033	0.270		
	4	0.108	0.123	0.023	0.184		
	5	0.108	0.124	0.017	0.135		
	6	0.125	0.060	0.518	8.660		
5	1	0.280	0.290	0.214	0.740	0.1291	34
	2	0.280	0.291	0.205	0.702		
	3	0.107	0.118	0.033	0.279		
	4	0.107	0.119	0.023	0.192		
	5	0.107	0.120	0.017	0.142		
	6	0.120	0.062	0.508	8.140		

increased: in addition to n (VLE) or 2n (LLE) activity coefficient evaluations, n^2 (VLE) or $2n^2$ (LLE) activity coefficient composition derivatives are needed. Of course, if some or all of the derivatives were not used, additional savings could be achieved. Rebeyrote (1980) and Stell(1981) excluded derivatives completely for their ideal VLE systems. Bryan(1985) ,in analyzing the selective partial use of derivatives in his LLE flash calculations, found that virtually all his systems required derivatives. In this work, the complete n x n matrix of derivatives was used for all systems.

Michelsen (1982) concluded from his study of VLE calculations that it was often desirable to begin the NR process after a small number of iterations from a first-order method. If the NR method is implemented with relatively poor initial estimates, oscillations in the iteration variables are often observed, and the convergence is not monotonic (Bryan, 1985). Second order techniques often display this behavior when far from convergence. All techniques utilizing the NR algorithm were therefore initialized with three SS iterations.

The results of these calculations in terms of number of MT evaluations are presented in tables 12 - 13. The major drawback of this method is the extra cost of using

Table 12. Costs in Terms of MT Evaluations for The NR Method (VLE mixtures)

System	Ex. #	Initial SS Iterations	NR Iterations	Total MT Evaluations
1	1	3	5	69
	2	3	4	57
	3	3	4	57
	4	3	4	57
	5	3	4	57
2	1	3	12	153
	2	3	9	117
	3	3	7	93
	4	3	6	81
	5	3	5	69
3	1	3	4	92
	2	3	4	92
	3	3	5	112
	4	3	8	342
	5	3	5	112
4	1	3	5	112
	2	3	5	112
	3	3	5	112
	4	3	5	112
	5	3	5	112
5	1	3	5	228
	2	3	6	270
	3	3	6	270
	4	3	6	270
	5	3	6	270

Table 13. Costs in Terms of MT Evaluations for The NR Method(LLE mixtures)

System	Ex. #	Initial SS Iterations	NR Iterations	Total MT Evaluations
6	1	3	3	90
	2	3	4	114
	3	3	4	114
	4	3	4	114
	5	3	4	114
7	1	3	4	114
	2	3	4	114
	3	3	5	138
	4	3	5	138
	5	3	5	138
8	1	3	4	184
	2	3	4	184
	3	3	4	184
	4	3	5	224
	5	3	5	224
9	1	3	4	184
	2	3	4	184
	3	3	4	184
	4	3	5	224
	5	3	5	224
10	1	3	3	288
	2	3	4	372
	3	3	4	372
	4	3	5	456
	5	3	5	456

derivatives. One way of improving the NR technique is to replace the underlying MT model (NRTL) with a simple analytic correlation that will be created from MT data and used to provide properties at less computational cost.

Convergence of VLE and LLE Flash Calculations by The Local Approximation Technique

If the MT model (NRTL) can be replaced by a simpler correlation for most of the iterative calculation, the computational savings will be substantial. The pseudo-ternary (equation 32), pseudo-quarternary (equation 39), expanded-pseudo-ternary (equation 40), and expanded-pseudo-quarternary (equation 41) two-suffix Margules equations were described in chapter II. These four LA models will be implemented into the NR convergence scheme in such a way that the number of property evaluations usually required of the underlying MT model will be decreased.

Each of the models is linear in its parameters, and this allows for the use of a computationally simple regression scheme i.e. sophisticated non-linear parameter fitting techniques can be avoided. A real mixture is first converted into a pseudo-ternary or pseudo-quarternary system in which the real components have been combined according to the

definitions presented in chapter II. A local approximation is then set up for each component in the system. Consider an example LLE mixture of three components: six models, representing each component in each phase are then created. Each model represents the component of interest as a function of the two (ternary models) or three (quarternary models) pseudo-components.

The model must be capable of providing accurate property values, and so its creation is an important step. The LA model, as its name indicates, applies to a local area that can be characterized by its thermodynamic state-space (TSS) (Rebeyrote, 1980). This phrase is simply defined as the properties that describe a thermodynamic state. In this work, the TSS comprises T, P, x_i and y_i. The LA models are simple functions of composition and, therefore, compositions must be used to fit the parameters. After MT activity coefficients are obtained from these compositions, the parameter regression can begin (see Appendix B). The mathematics of linear regression require at least as many points as there are parameters. Bryan (1985), Stell (1981), and Rebeyrote (1980) all found that the minimum number of points was never enough to create an equation capable of extapolating accurate properties. Bryan (1985) used five composition sets (data points) to fit his three LA model

parameters and devised a standard method for their selection. When he chose to create or update his model in an LLE flash calculation, he obtained the mixture mole fractions at the current iteration and reduced them to pseudo-ternary compositions (when the number of components was equal to three, no reduction was necesssary). He then perturbed this base point on a ternary diagram to obtain four other data points on a semicircle surrounding it. The data points are then converted back to real compositions and used in the regression. The 'cost' (in terms of MT evaluations) of creating his LA model was always twice the number of components times five (2*n*5). The details of this procedure are presented in Appendix J. This scheme was attempted in this work and failed for all pseudo-quartenary models. The regressions for the parameters of these models involved inversion of matrices that were singular or nearly singular. This occurs when supposedly independent variables are actually dependent.

The point selection scheme used in this work does not collapse a 'real composition set' into a 'pseudo-set'. Given a data set from the computational procedure, the following steps are taken:

1) Consider component one in this n component set. If its value is greater than .5, multiply it by

1.1 and renormalize . Now, multiply the component by .9 and renormalize again.

2) Repeat this procedure for all n components.
3) Going through each component again, find the ones that have compositions below .1 . Renormalize this component's set twice after multiplying it by 1.6 and .4
4) Similar perturbations are performed for components whose compositions lie between .5 and .1.

The numbers used to multiply the compositions are chosen carefully so data sets far removed from the base point or data sets virtually identical to the base point are avoided e.g. if a component has a high value, a factor very close to one will be multiplied by it and the resulting normalized point will not differ dramatically from the original set. If the points are widely scattered and are used in the parameter regresssion, then the model's applicability becomes more global than local. The parameters of every LA model are regressed by this point selection scheme. For every n component system, 2n points will be created in addition to the base point i.e. 2n+1 points. Appendix K lists some examples of the highly accurate activity coefficients and good derivative estimates provided by these

models.

Results of VLE and LLE Flash Calculations By the LA Method

All the LA models were implemented with the NR algorithm where three SS iterations were used to initialize the process variables. The third iteration provides the base point that is used in the point selection scheme. Linear-least squares regression then uses these phase compositions and NRTL activity coefficients to fit the LA model parameters.

Equations (32) and (39) may be called LA1 and LA2 respectively. After they are created, they completely replace the MT model and provide both activity coefficients and derivatives until they have converged to their solutions. At this point, one has the choice of using the 'full' NRTL to converge to the 'true' solution. This, however, will tend to be costly due to the need for derivatives from a sophisticated analytic expression. In this work, after the LA 'solution' has been found, the LA model is used to provide derivatives through each iteration. The NRTL provides activity coefficients, and when the objective function is minimized to the appropriate tolerance (5×10^{-5}), the 'real' solution is obtained. Tables 14 - 17

list the number of MT evaluations required to converge the various example flashes.

Equations (40) and (41) are created in the same way that equations (32) and (39) are. After they have converged to their solution, the base points around which they were created (see chapter II) are updated with these new values. Activity coefficients from the NRTL model are also provided at an update. The models are therefore perfect i.e. the activity coefficients they provide will be exactly equal to the NRTL values. The objective function is checked at this stage, and if it is sufficiently minimized, the 'true' solution has been reached. If not, the procedure continues with further updates and further objective function checks. The pseudo-ternary and pseudo-quarternary systems are designated by LA3 and LA4 respectively. The MT evaluations required of the LA3 model are shown in tables 18 and 19. The cumbersome nine parameter LA4 did not converge with any of the example flashes. Its coefficient matrix, $\mathbf{X}^T\mathbf{X}$ (see Appendix J) is unstable and displays singularity . Lapidus (1963), in his discussion of parameter regression, concluded that when more than six parameters are involved in a least-squares regression, the technique will often fail.

The expansion forms can be truncated to two and three parameter linear equations. Let us call these LA5(a) and

Table 14. Costs in Terms of MT Evaluations for The LA1 Method (VLE mixtures)

System	Ex. #	Initial SS Iterations	Iteration # Which LA Converges	Total LA Iterations	Total MT Evaluations
1	1	3	2	6	42
	2	3	3	10	54
	3	3	3	13	60
	4	3	3	17	72
	5	3	3	23	90
2	1	3	3	12	57
	2	3	3	13	60
	3	3	3	14	63
	4	3	3	17	72
	5	3	2	22	87
3	1	3	2	6	64
	2	3	2	6	64
	3	3	3	10	80
	4	3	3	13	88
	5	3	2	17	104
4	1	3	3	8	72
	2	3	3	11	80
	3	3	3	8	68
	4	3	3	14	92
	5	3	2	12	84
5	1	3	2	8	132
	2	3	2	8	132
	3	3	2	9	138
	4	3	2	10	144
	5	3	2	10	144

Table 15. Costs in Terms of MT Evaluations for The LA1 Method (LLE mixtures)

System	Ex. #	Initial SS Iterations	Iteration # Which LA Converges	Total LA Iterations	Total MT Evaluations
6	1	3	3	9	102
	2	3	3	13	120
	3	3	4	17	144
	4	3	6	15	126
	5	3	2	29	198
7	1	3	3	7	90
	2	3	4	9	96
	3	3	5	14	120
	4	3	8	62	402
	5	3	2	75	462
8	1	3	2	10	160
	2	3	2	13	184
	3	3	2	14	192
	4	3	3	16	208
	5	3	2	19	224
9	1	3	2	10	160
	2	3	2	10	160
	3	3	3	11	168
	4	3	3	13	176
	5	3	3	15	192
10	1	3	3	7	252
	2	3	3	8	264
	3	3	4	11	288
	4	3	4	12	288
	5	3	2	12	288

Table 16. Costs in Terms of MT Evaluations for The LA2 Method (VLE mixtures)

System	Ex. #	Initial SS Iterations	LA Gamma LA Derivs. Iterations	Total # of LA Iterations	Total MT Evaluations
3	1	3	2	6	64
	2	3	3	5	60
	3	3	4	12	84
	4	3	4	16	96
	5	3	2	20	112
4	1	3	3	9	76
	2	3	3	9	72
	3	3	3	9	72
	4	3	3	14	92
	5	3	2	12	84
5	1	3	2	8	132
	2	3	2	8	132
	3	3	3	7	126
	4	3	3	8	126
	5	3	2	10	138

Table 17. Costs in Terms of MT Evaluations for The LA2 Method (LLE mixtures)

System	Ex. #	Initial SS Iterations	Iteration # Which LA Converges	Total LA Iterations	Total MT Evaluations
8	1	3	3	7	136
	2	3	3	9	144
	3	3	3	10	152
	4	3	4	10	152
	5	3	2	12	160
9	1	3	2	9	152
	2	3	2	9	152
	3	3	3	10	160
	4	3	3	12	168
	5	3	2	12	168
10	1	3	3	7	252
	2	3	3	9	264
	3	3	4	9	264
	4	3	5	15	324
	5	3	2	16	324

Table 18. Costs in Terms of MT Evaluations for The LA3 Method (VLE mixtures)

System	Ex. #	Initial SS Iterations	Total LA Iterations	Number of Updates	Total MT Evaluations
1	1	3	5	2	54
	2	3	5	2	54
	3	3	6	2	54
	4	3	8	3	57
	5	3	11	4	60
2	1	3	6	2	54
	2	3	6	2	54
	3	3	6	2	54
	4	3	10	3	57
	5	3	9	3	57
3	1	3	5	3	56
	2	3	7	3	60
	3	3	13	5	72
	4	3	15	4	68
	5	3	12	4	76
4	1	3	12	5	68
	2	3	6	2	56
	3	3	15	6	72
	4	3	14	5	68
	5	3	18	7	76
5	1	3	10	4	120
	2	3	13	5	126
	3	3	11	4	120
	4	3	11	4	120
	5	3	8	3	114

Table 19. Costs in Terms of MT Evaluations for The LA3 Method (LLE mixtures)

System	Ex. #	Initial SS Iterations	Total LA Iterations	Number of Updates	Total MT Evaluations
6	1	3	5	2	108
	2	3	5	2	108
	3	3	5	2	108
	4	3	6	2	108
	5	3	6	2	108
7	1	3	5	2	108
	2	3	6	2	108
	3	3	6	2	108
	4	3	9	3	114
	5	3	9	3	114
8	1	3	54	24	288
9	1	3	10	4	128
	2	3	13	5	136
	3	3	13	5	136
	4	3	22	9	168
10	1	3	9	4	240
	2	3	12	5	252
	3	3	14	5	252
	4	3	16	6	264
	5	3	19	7	276

LA6(a) respectively. When these are implemented in the exact same way as their predecessors, tables 20 -23 show that they converge quite effectively.

Comparison of the Computational Efficiencies of the LA Models

Tables 24 - 25 compare the MT evaluations that each of the NR/LA methods achieved. The 'straight' SS and NR approaches tend to use the most MT evaluations. When models LA1, LA3 and LA5(a) are compared in terms of the total average MT evaluations for all the LLE flashes, the increase in savings from the NR method to the LA5(a) is approximately 49%. For the VLE flashes, the savings between the higher SS and the LA5(a) is about 40%. The LA3 model also displays marked savings. When the pseudo-quarternary models are used, there is no point in applying them to ternary systems because they will simply collapse to a pseudo-ternary. When these models (LA2, LA6(a)) are compared to the rest, only the three systems with components greater than three were considered. The MT evaluations were then averaged over three mixtures for each phase. In the VLE systems, the savings between the highest and lowest values were of the order of 50%. In the LLE systems, LA6(a)

Table 20. Costs in Terms of MT Evaluations for The LA5(a) Method (VLE mixtures)

System	Ex. #	Initial SS Iterations	Total LA Iterations	Number of Updates	Total MT Evaluations
1	1	3	6	2	36
	2	3	10	4	42
	3	3	15	6	48
	4	3	23	9	57
	5	3	35	13	69
2	1	3	12	4	42
	2	3	12	4	42
	3	3	11	4	42
	4	3	17	6	48
	5	3	28	11	63
3	1	3	5	2	56
	2	3	8	3	60
	3	3	10	4	64
	4	3	33	11	92
	5	3	27	10	88
4	1	3	12	5	68
	2	3	10	4	64
	3	3	14	5	68
	4	3	20	7	76
	5	3	13	4	64
5	1	3	7	3	114
	2	3	9	3	114
	3	3	9	3	114
	4	3	8	3	114
	5	3	8	3	114

Table 21. Costs in Terms of MT Evaluations for The LA5(a) Method (LLE mixtures)

System	Ex. #	Initial SS Iterations	Total LA Iterations	Number of Updates	Total MT Evaluations
6	1	3	5	2	72
	2	3	5	2	72
	3	3	5	2	72
	4	3	8	3	78
	5	3	27	11	126
7	1	3	7	3	78
	2	3	8	3	78
	3	3	10	4	84
	4	3	11	4	84
	5	3	11	4	84
8	1	3	16	7	152
	2	3	20	8	160
	3	3	20	8	160
	4	3	21	8	160
	5	3	36	13	200
9	1	3	10	4	128
	2	3	13	5	136
	3	3	14	5	136
10	1	3	6	4	240
	2	3	12	5	252
	3	3	14	5	252
	4	3	17	6	264
	5	3	17	6	264

Table 22. Costs in Terms of MT Evaluations for The LA6(a) Method (VLE mixtures)

System	Ex. #	Initial SS Iterations	Total LA Iterations	Number of Updates	Total MT Evaluations
3	1	3	5	2	56
	2	3	6	2	56
	3	3	11	4	64
	4	3	33	11	92
	5	3	25	9	84
4	1	3	5	2	56
	2	3	8	3	60
	3	3	12	5	68
	4	3	20	7	76
	5	3	17	6	72
5	1	3	5	2	108
	2	3	6	2	108
	3	3	8	3	114
	4	3	8	3	114
	5	3	10	4	120

Table 23. Costs in Terms of MT Evaluations for The LA6(a) Method (LLE mixtures)

System	Ex. #	Initial SS Iterations	Total LA Iterations	Number of Updates	Total MT Evaluations
8	1	3	5	2	112
	2	3	8	4	128
	3	3	13	5	136
	4	3	40	13	200
9	1	3	11	4	128
	2	3	16	6	144
	3	3	46	16	224
10	1	3	7	3	228
	2	3	9	3	228
	3	3	11	4	240
	4	3	13	4	240

Table 24. Computational Efficiencies of The SS, NR, and LA Convergence Methods (VLE)

System	Ex. #	Total Number of Evaluations (SS)	(NR)	(LA1)	(LA2)	(LA3)	(LA5)	(LA6)
1	1	51	69	42	-	54	36	-
	2	66	57	54	-	54	42	-
	3	84	57	60	-	54	48	-
	4	114	57	72	-	57	57	-
	5	159	57	90	-	60	69	-
2	1	51	153	57	-	54	42	-
	2	63	117	60	-	54	42	-
	3	79	93	63	-	54	42	-
	4	87	81	72	-	57	48	-
	5	126	69	87	-	57	63	-
3	1	36	92	64	64	56	56	56
	2	64	92	64	60	60	60	56
	3	88	112	80	84	68	64	64
	4	132	342	88	96	64	92	92
	5	148	112	104	112	64	88	84
4	1	56	112	72	76	68	68	56
	2	58	112	80	72	56	64	60
	3	96	112	68	72	72	68	68
	4	104	112	92	92	68	76	76
	5	108	112	84	84	76	64	72
5	1	90	228	132	132	120	114	108
	2	96	270	132	132	126	114	108
	3	102	270	138	126	120	114	114
	4	108	270	144	126	120	114	114
	5	114	270	144	138	114	114	120

Table 25. Computational Efficiencies of the SS, NR, and LA Convergence Methods (LLE)

System	Ex. #	Total Number of MT Evaluations (SS)	(NR)	(LA1)	(LA2)	(LA3)	(LA5(a))	(LA6(a))
6	1	84	90	102	-	108	72	-
	2	96	114	120	-	108	72	-
	3	114	114	144	-	108	72	-
	4	188	114	126	-	108	78	-
	5	186	114	198	-	108	126	-
7	1	102	114	90	-	108	78	-
	2	138	114	96	-	108	78	-
	3	186	138	120	-	108	84	-
	4	270	138	402	-	114	84	-
	5	330	138	462	-	114	84	-
8	1	144	184	160	136	288	152	112
	2	232	184	184	144	nc	160	128
	3	256	184	192	152	nc	160	136
	4	368	224	208	152	nc	160	200
	5	544	224	224	160	nc	200	nc
9	1	184	184	160	152	128	128	128
	2	224	184	160	152	136	136	144
	3	264	184	168	160	136	136	224
	4	328	224	176	168	168	nc	nc
	5	392	224	192	168	nc	nc	nc
10	1	132	288	252	252	240	240	228
	2	204	372	264	264	252	252	228
	3	312	372	288	264	252	252	240
	4	384	456	288	324	264	264	240
	5	408	456	288	324	276	264	nc

showed a 37 % improvement over the SS technique.

The savings are seen to be substantial but could be easily improved. If we take the two and three parameter equations and fit them to fewer points (identify them as LA5(b) and LA6(b) respectively), savings are seen to be achieved without loss of accuracy (see Appendix K) . The first six points provided by the point selection scheme are used in the regression. Tables 26 - 29 list the MT evaluations of these models. Tables 30 and 31 compare the efficiencies of the models when more or less points are used in the regression scheme i.e between LA5(a) and LA6(b) and LA6(a) and LA6(b). By comparing average MT evaluations, we see that an additional savings has occurred.

Table 26. Costs in Terms of MT Evaluations for The LA5(b) Method (VLE mixtures)

System	Ex. #	Initial SS Iterations	Total LA Iterations	Number of Updates	Total MT Evaluations
1	1	3	8	3	36
	2	3	10	4	39
	3	3	15	6	45
	4	3	23	9	54
	5	3	34	13	69
2	1	3	13	5	42
	2	3	16	6	45
	3	3	18	7	48
	4	3	23	9	54
	5	3	34	14	69
3	1	3	5	2	44
	2	3	9	4	52
	3	3	13	5	66
	4	3	32	11	80
	5	3	28	10	76
4	1	3	7	3	44
	2	3	13	5	52
	3	3	13	5	66
	4	3	23	8	80
	5	3	14	5	66
5	1	3	13	5	84
	2	3	16	6	90
	3	3	13	5	84
	4	3	12	4	78
	5	3	10	3	72

Table 27. Costs in Terms of MT Evaluations for The LA5(b) Method (LLE mixtures)

System	Ex. #	Initial SS Iterations	Total LA Iterations	Number of Updates	Total MT Evaluations
6	1	3	7	3	72
	2	3	7	3	72
	3	3	8	3	72
	4	3	9	3	72
	5	3	9	3	72
7	1	3	7	3	72
	2	3	8	3	72
	3	3	8	3	72
	4	3	11	4	78
	5	3	11	4	78
8	1	3	16	7	128
	2	3	18	7	128
	3	3	16	9	120
	4	3	23	9	144
9	1	3	12	5	112
	2	3	13	5	112
	3	3	12	4	104
	4	3	20	7	128
10	1	3	9	4	156
	2	3	9	5	168
	3	3	16	6	180
	4	3	15	5	168
	5	3	15	5	168

Table 28. Costs in Terms of MT Evaluations for The LA6(b) Method (VLE mixtures)

System	Ex. #	Initial SS Iterations	Total LA Iterations	Number of Updates	Total MT Evaluations
3	1	3	5	2	40
	2	3	10	4	48
	3	3	13	5	52
	4	3	32	11	76
	5	3	28	10	72
4	1	3	7	3	44
	2	3	13	5	52
	3	3	18	7	60
	4	3	23	8	64
	5	3	18	7	60
5	1	3	10	4	72
	2	3	10	4	72
	3	3	14	5	78
	4	3	14	5	78

Table 29. Costs in Terms of MT Evaluations for The LA6(b) Method (LLE)

System	Ex. #	Initial SS Iterations	Total LA Iterations	Number of Updates	Total MT Evaluations
8	1	3	7	3	88
	2	3	8	3	88
	3	3	8	3	88
	4	3	23	8	128
9	1	3	11	4	96
	2	3	8	3	88
	3	3	8	3	88
	4	3	23	8	128
10	1	3	27	10	216
	2	3	18	7	180
	3	3	21	7	180

Table 30. Computational Efficiencies of The LA5(a), LA5(b),LA6(a), and LA6(b) Models (VLE)

System	Ex. #	Total Number of MT Evaluations LA5(a)	LA5(b)	LA6(a)	LA6(b)
1	1	36	36	-	-
	2	42	39	-	-
	3	48	45	-	-
	4	57	54	-	-
	5	69	66	-	-
2	1	42	42	-	-
	2	42	45	-	-
	3	42	48	-	-
	4	48	54	-	-
	5	63	69	-	-
3	1	56	44	56	40
	2	60	52	56	48
	3	64	66	64	52
	4	92	80	92	76
	5	88	76	84	72
4	1	68	48	56	44
	2	64	66	60	52
	3	68	66	68	60
	4	76	68	76	64
	5	64	66	72	60
5	1	114	84	108	72
	2	114	90	108	72
	3	114	84	114	78
	4	114	78	114	78
	5	114	72	120	

Table 31. Computational Efficiencies of the LA5(a), LA5(b), LA6(a), and LA6(b) Models (LLE)

System	Ex. #	Total Number of MT Evaluations LA5(a)	LA5(b)	LA6(a)	LA6(b)
6	1	72	72	-	-
	2	72	72	-	-
	3	72	72	-	-
	4	78	72	-	-
	5	126	72	-	-
7	1	78	72	-	-
	2	78	72	-	-
	3	84	72	-	-
	4	84	78	-	-
	5	84	78	-	-
8	1	152	128	112	88
	2	160	128	128	88
	3	160	120	136	88
	4	160	144	200	128
	5	160	nc	nc	nc
9	1	128	112	128	96
	2	136	112	144	88
	3	136	104	224	88
	4	nc	128	nc	128
	5	nc	nc	nc	nc
10	1	240	156	228	216
	2	252	168	228	180
	3	252	180	240	180
	4	264	168	240	nc
	5	264	168	nc	nc

IV-CONCLUSION

The application of local approximations for thermodynamic properties in single-stage flash calculations has been investigated. These simple expressions locally replace evaluations of complicated molecular thermodynamic models to achieve reductions in computational effort. This work has concentrated on analyzing different forms and different implementations of the local approximations in the computational scheme. These models were fitted to a small number of data points from the underlying thermodynamic relation and then used to replace that model whenever possible.

Pseudo-ternary and pseudo-quarternary forms of the two-suffix Margules equations were found to give adequate approximation of the behavior of non-ideal liquid systems. Their greatest asset was their ability to predict composition derivatives of activity - coefficients. This enabled the models to be used with highly efficient computational methods such as the multivariable Newton-Raphson convergence technique.

Reductions in the number of molecular thermodynamic evaluations by as much as 55% were achieved. In multiple stage extraction or distillation , the savings would be

expected to be much more.

The local approximation method can be applied to a number of challenging problems like dynamic and process control simulations and flowsheet calculations. In a flowsheet environment, analytical expressions for partial derivatives are simply not available. When they are made available by perturbation, they are usually prohibitively expensive to calculate. Approximation then becomes a viable alternative.

Bibliography

Ammar, M.N. and Renon, H. 1987. AIChE Journal. 33: 926-939.

Barrett, A. and Walsh, J.J. 1979. Computers and Chemical Engineering. 3: 397-402.

Biegler, A., Grossman, I.E. and Westerberg, A.W. 1984. Computers and Chemical Engineering. 9: 201-206.

Boston, J.F. and Britt, H.I. 1978. Computers and Chemical Engineering. 2: 109-122.

Brian, P.L.T. 1972. Staged Cascades in Chemical Processing. Englewood-Cliffs : Prentice-Hall.

Broyden, C.G. 1965. Math. Comp. 19: 577.

Bryan, P.B. 1985. Representation of Thermodynamic Properties by Local Approximations in Liquid-Liquid Equilibrium. Ph.D dissertation, University of California, Berkeley.

Grens, E.A. 1983. " Efficient Use of Thermodynamic Models in Process Calculations. " Proc. 2nd Int. Conf. Foundations Computer-Aided Process Design, CACHE. : 249.

Govindaswamy, S., Andiappan, AN., Lakshmanan, SM. 1977. Journal of Chemical and Engineering Data. 22: 264-269.

Griswold, J. and Buford, C.B. 1949. Industrial and Engineering Chemistry. 41: 2347-2351.

Heidemann, R.A. 1983. Fluid Phase Equilibria. 14: 55-78.

Henley, E.J. and Rosen, E.M. 1969. Material and Energy Balance Computations. New York: Wiley.

Henley, E.J. and Seader, J.D. 1981. Equilibrium Stage Separation Operations in Chemical Engineering. New York: John Wiley & Sons.

Hillestad, M. and Hertzberg, T. 1985. Computers and Chemical Engineering. 4: 377-388.

Holland, C.D. 1975. Fundamentals and Modelling of

Separation Processes. Englewood-Cliffs : Prentice-Hall.

Joulia, X., Massiochi, P., Koehret, B., Paradowski, H. and Bartuel, J.J. 1986. Fluid Phase Equilibria. 26: 15-36.

Joy, D.S. and Kyle, B.G. 1969. AIChE Journal. 15: 298-300.

Kehat, E. and Ghitis, B. 1981. Comp. & Chem. Eng. 5: 171.

Kikic, I., Alessi, P., Colussi, I. , Orlandini, M. 1982. The Canadian Journal of Chemical Engineering. 60: 168-171.

King, C.J. 1980. Separation Processes. 2d Ed. New York: Mcgraw-Hill

Kubicek, M., Hlavacek, V. and Prochaska, F. 1975. Chemical Engineering Science. 31: 277-284.

Lapidus, L. 1962. Digital Computations for Chemical Engineers. New York: Mcgraw-Hill.

Leesley, M.E. and Heyen, G. 1976. Computers and Chemical Engineering. 1: 109-112.

Lucia, A. and Machietto, S. 1983. AIChE Journal. 29: 705-712.

Lucia, A., Miller, D.C., Ashwani, K. 1985. AIChE Journal. 31: 1381-1388.

Lucia, A. 1985. AIChE Journal. 31: 558-566.

Macchietto, S., Chimowitz, E.H. and Anderson, T.F. 1984. AIChE Journal.30: 669-672.

Mah, R.S.H. and Lin, T.D. 1980. Computers and Chemical Engineering.4: 75-78.

Michelsen, M.L. 1981. "Partial Derivatives of Thermodynamic Properties from Equations of State." Report SEP 8109, Instituttet for Kemiteknik, Technical University of Denmark.

Michelsen, M.L. 1981. Fluid Phase Equilibria.9: 21-40.

Michelsen, M.L., Mollerup, J. 1986. AIChE Journal.32: 1389-1392.

Nagata, J. 1973. J. Chem. Engng. Japan.6: 18.

Napthali, L.M. and Sandholm, D.P. 1971. AIChE Journal.17: 148-153.

Ohanomah, M.O. 1981. Computational Algorithms for Multicomponent Phase Equiibria and Distillation. Ph.D dissertation, Univ. of British Columbia, Vancouver.

Ohanomah, M.O. and Thompson, D.W. 1983. Computers and Chemical Engineering.8: 147-156.

Ohanomah, M.O. and Thompson, D.W. 1983. Computers and Chemical Engineering.8: 157-162.

Ohanomah, M.O. and Thompson, D.W. 1983. Computers and Chemical Engineering.8: 163-180.

Prausnitz, J.M., Anderson, T.F., Grens, E.A., Eckert, C.A., Hsieh, R. and O'connel, J.P. 1980. Computer Calculations for Multicomponent Vapor-liquid and Liquid-liquid Equilibria. Englewood-cliffs: Prentice-Hall.

Prausnitz, J.M., Lichtenthaler, R.N. and Azevedo, E.G. 1985. Molecular Thermodynamics of Fluid-Phase Equilibria. Englewood-Cliffs: Prentice-Hall.

Rachford, H.H. and Rice, J.D. 1952. J. Petrol. Technol.1: 19.

Rebeyrote, G. 1980. Use of Local Approximations for Thermodynamic Properties in Staged Separation Calculations. Master's Thesis, University of Cal., Berkeley.

Reid, R.C., Sherwood, and Prausnitz, J.M. 1977. The Properties of Gases and Liquids. New York: Mcgraw-Hill.

Renon, H. and Prausnitz, J.M. 1968. AIChE Journal.14: 135-144.

Renon, H., Asselineau, L., Cohen, G., and Raimbault, C. 1971. Calcul Sur Ordinateur Des Equilibres Liquide -vapeur et Liquide-Liquide. Paris; Editions Technip.

Schubert, L.K. 1970. Math. Comp.24: 27-30.

Sereno, A.M., Anderson, T.F. and Medina, A.G. 1986. Computers and Chemical Engineering.11: 177-185.

Shacham, M., Machietto, S., Stutzman, L.F. and Babcock, P. 1981. Computers and Chemical Engineering.6: 79-95.

Smith, J.M. and Van Ness, H.C. 1975. Introduction to Chemical Engineering Thermodynamics. New York: McGraw -Hill.

Stell, M.F. 1979. Efficient Application of Local Approximations for Thermodynamic Properties in Multicomponent Distillation Calculations. Master's Thesis, University of Cal., Berkeley.

Treybal, R.E. 1968. Mass Transfer Operations. New York: McGraw-Hill.

Tyreus, B.D., Luyben, W.L. and Schlesser, W.E. 1975. Ind. Eng. Chem. Process Des. Dev.14: 427-432.

Venkataraman, S. and Lucia, A. 1987. Computers and Chemical Engineering.12: 55-69.

Venkataraman, S. and Lucia, A. 1986. AIChE Jounal.32: 1057-1066.

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

Wegstein, J.H. 1958. Commun. A.C.M.9:9.

Westman, K.R., Lucia, A. and Miller, D.C. 1984. Computers and Chemical Engineering.8: 219-228

Appendix A

The NRTL Equation for Multicomponent Mixtures

The NRTL equation of the excess Gibbs energy for multicomponent mixtures is expressed entirely in terms of binary parameters. Gibbs energies of interaction between molecules $_i$ and $_j$ are represented by g_{ij} ($g_{ij} = g_{ji}$ and g_{ii} are the Gibbs energies of the pure components), and liquid mole fractions of the n components are defined by x_n. The interaction parameters can be grouped as follows:

$$\tau_{ji} = \frac{\left(g_{ji} - g_{ii} \right)}{RT} \qquad \text{(A-1)}$$

where $\tau_{ii} = \tau_{jj} = 0$. A parameter G_{ji} may also be defined as:

$$G_{ji} = \exp(-\alpha_{ji}\tau_{ji}) \qquad \text{(A-2)}$$

where $G_{ii} = G_{jj}$ and α_{ji} is an empirical constant characterizing the nonrandomness of the mixture. The NRTL activity coefficient correlation can now be expressed as:

$$\ln\gamma_i = \frac{\sum_{j=1}^{m} \tau_{ji} G_{ji} x_j}{\sum_{l=1}^{m} G_{li} x_l} + \sum_{j=1}^{m} \frac{x_j G_{ij}}{\sum_{l=1}^{m} G_{lj} x_l} \left(\tau_{ij} - \frac{\sum_{n=1}^{m} x_n \tau_{nj} G_{nj}}{\sum_{l=1}^{m} G_{lj} x_l} \right) \qquad \text{(A-3)}$$

Appendix B

Mathematical Details of Parameter Regression

The parameters of the local approximation models can be evaluated by least-squares regression. The pseudo-ternary two-suffix Margules equation is:

$$RT\ln\gamma_i = A_i x_B^2 + B_i x_C^2 + C_i x_B x_C \qquad \text{(B-1)}$$

This can be expressed as:

$$y_i = A_i z_1 + B_i z_2 + C_i z_3 \qquad \text{(B-2)}$$

where $y_i = RT\ln\gamma_i$ and the z_i terms represent the independent variables made up of the pseudo-mole fractions. At least three equations (created from three data sets) will be needed to solve for the parameters. The resulting system of equations in matrix notation is:

$$\mathbf{Y} = \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ y_n \end{bmatrix} \quad ; \quad \mathbf{X} = \begin{bmatrix} 0 & z_{11} & z_{12} & z_{13} \\ 0 & z_{21} & z_{22} & z_{23} \\ . & . & . & . \\ . & . & . & . \\ 0 & z_{n1} & z_{n2} & z_{n3} \end{bmatrix} \quad ; \quad \mathbf{P} = \begin{bmatrix} A \\ B \\ C \end{bmatrix} \qquad \text{(B-3)}$$

The regression equation :

$$\mathbf{Y} = \mathbf{XP} \tag{B-4}$$

can now be expressed in least squares normal form:

$$(\mathbf{X}^T\mathbf{X})\,\mathbf{P} = \mathbf{X}^T\mathbf{Y} \tag{B-5}$$

where $\mathbf{X}^T$ is the transpose of $\mathbf{X}$. For M data sets, equation (B-2) can be represented in this form by:

$$A_i \sum_1^M z_1^2 + B_i \sum_1^M z_1 z_2 + C_i \sum_1^M z_1 z_3 = \sum_1^M yz_1 \tag{B-6}$$

$$A_i \sum_1^M z_1 z_2 + B_i \sum_1^M z_2^2 + C_i \sum_1^M z_{23} = \sum_1^M yz_2 \tag{B-7}$$

$$A_i \sum_1^M z_3 z_1 + B_i \sum_1^M z_3 z_2 + C_i \sum_1^M z_3^2 = \sum_1^M yz_3 \tag{B-8}$$

The vector $\mathbf{P}$ can now be calculated as:

$$\mathbf{P} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

where $(\mathbf{X}^T\mathbf{X})^{-1}$ is the inverse of $(\mathbf{X}^T\mathbf{X})$.

Appendix C

NRTL parameters of Example Systems

The NRTL equation has three adjustable parameters for each binary pair in the multicomponent system: c_{ij}, c_{ji} and α_{ij}, where i and j are the component indices. The c_{ij} have units of cal/mole and the α_{ij} are dimensionless. The parameters of five LLE systems and five VLE systems are listed and were obtained from the following sources:

(a) Renon et al (1971)

(b) Nagata (1972)

(c) Kikic et al (1982)

(d) Muhopadhyay and Sahasranaman (1982)

Acetonitrile(1) Methanol(2) n-Octane(3)

Type(LLE) Conditions(T=25C, P=760mmHg) Source(d)

i-j	C_{ij}	C_{ji}	α_{ij}
1-2	-736.52	463.43	.20
1-3	1572.87	1138.63	.20
2-3	1285.83	111.09	.20

DMF(1) Water(2) Benzene(3)

Type(LLE) Conditions(T=20C, P=760mmHg) Source(a)

i-j	C_{ij}	C_{ji}	α_{ij}
1-2	-1240	1460	.253
1-3	394	-140	.425
2-3	3350	2120	.203

n-heptane(1) DMF(2) Water(3) Benzene(4)

Type(LLE) Conditions(T=20C, P=760mmHg) Source(a)

i-j	C_{ij}	C_{ji}	α_{ij}
1-2	1113	1186	.25
1-3	2800	4100	.15
1-4	-207	697	.30
2-3	-1240	1460	.253
2-4	394	-140	.425
3-4	3350	2120	.203

Iso-octane(1) Cyclohexane(2) Benzene(3) Furfural(4)

Type(LLE) Conditions(T=20C, P=760mmHg) Source(a)

i-j	C_{ij}	C_{ji}	α_{ij}
1-2	-262	428	.30
1-3	-207	824	.30
1-4	1284	1400	.30
2-3	-37	445	.30
2-4	1420	979	.35
3-4	434	197	.30

Benzene(1) Toluene(2)
Cyclohexane(3) Hexane(4) Heptane(5) Sulfolane(6)
Type(LLE) Conditions(T=75C, P=760mmHg) Source(c)

i-j	c_{ij}	c_{ji}	α_{ij}
1-2	535.38	-447.38	.30
1-3	107.2	138.18	.30
1-4	667.2	-285.53	.30
1-5	823.8	-394.47	.30
1-6	1266.44	-149.30	.30
2-3	-435.11	978.06	.30
2-4	-467.90	913.13	.30
2-5	383.02	-109.31	.30
2-6	1733.81	-229.56	.30
3-4	296.23	-219.97	.30
3-5	-464.63	762.46	.30
3-6	2613.15	838.24	.20
4-5	-903.15	1349.01	.30
4-6	3396.61	1252.67	.20
5-6	2440.70	1609.99	.20

Hexane(1) Benzene(2) Ethanol(3)

Type(VLE) Conditions(T=60C, P=760mmHg) Source(a)

i-j	c_{ij}	c_{ji}	α_{ij}
1-2	-366.8	798	.2
1-3	1427.2	754.6	.3926
2-3	1082.4	420.8	.4142

Acetone(1) Methanol(2) Water(3)

Type(VLE) Conditions(T=63C, P=760mmHg) Source(b)

i-j	c_{ij}	c_{ji}	α_{ij}
1-2	-70	486	.3
1-3	200	1400	.3
2-3	-299	955	.3

Methanol(1) Ethanol(2) Acetone(3) Water(4)
Type(VLE) Conditions(T=65C, P=760mmHg) Source(b)

i-j	c_{ij}	c_{ji}	α_{ij}
1-2	-283	242	.30
1-3	486	-70	.30
1-4	-299	955	.30
2-3	540	-120	.30
2-4	-215	1488	.30
3-4	200	1400	.30

Cyclohexane(1) Benzene(2) Toluene(3) Ethanol(4)
Type(VLE) Conditions(T=70C, P=760mmHg) Source(a)

i-j	c_{ij}	c_{ji}	α_{ij}
1-2	-146.2	418.7	.2
1-3	139.6	86.6	.2
1-4	1479.6	842.4	.47
2-3	-737.8	984.2	.2
2-4	1040.3	407.6	.4048
3-4	954.3	758.0	.4945

Hexane(1) Heptane(2)
Cyclohexane(3) Benzene(4) Toluene(5) Ethanol(6)
Type(VLE) Conditions(T=70C, P=760mmHg) Source(a)

i-j	c_{ij}	c_{ji}	α_{ij}
1-2	-658.3	739.9	.20
1-3	-540.1	716.0	.20
1-4	-380.1	791.0	.20
1-5	-279.7	647.9	.20
1-6	1411.9	745.7	.3883
2-3	-297.5	301.1	.20
2-4	-869.0	1358.6	.1254
2-5	-251.6	547.4	.20
2-6	1458.5	1025.9	.4777
3-4	-146.2	418.7	.20
3-5	139.6	86.6	.20
3-6	1479.6	842.4	.4667
4-5	-737.8	984.2	.20
4-6	1040.3	407.6	.4048
5-6	954.3	758.0	.4945

Appendix D

Taylor Series Expansion of The Margules Equation

The pseudo-ternary two-suffix Margules equation may be written as:

$$RT\ln\gamma_1 = A(x_2^*)^2 + B(x_3^*)^2 + Cx_2^*x_3^* \qquad \text{(D-1)}$$

where 1, 2 and 3 represent the component indices and the x_i^* are mole fractions. In terms of mole numbers, x_i, equation (D-1) becomes:

$$RT\ln\gamma_1 = \frac{Ax_2^2}{\sigma^2} + \frac{Bx_3^2}{\sigma^2} + \frac{Cx_2x_3}{\sigma^2} \qquad \text{(D-2)}$$

where:

$$\sigma = x_1 + x_2 + x_3 \qquad \text{(D-3)}$$

In order to develop a Taylor series expansion, partial differentials of equation (D-2) with respect to mole numbers are needed:

$$\left(\frac{\partial f}{\partial x_1}\right)_{T,P,x_{j\neq 1}} = C1 \qquad \text{(D-4)}$$

$$\left(\frac{\partial f}{\partial x_2}\right)_{T,P,x_{j\neq 2}} = C2 \qquad \text{(D-5)}$$

$$\left(\frac{\partial f}{\partial x_3}\right)_{T,P,x_{j\neq 3}} = C3 \qquad \text{(D-6)}$$

$$.5\left(\frac{\partial^2 f}{\partial x_1^2}\right)_{T,P,x_{j\neq 1}} = C4 \qquad \text{(D-7)}$$

$$.5\left(\frac{\partial^2 f}{\partial x_2^2}\right)_{T,P,x_{j\neq 2}} = C5 \qquad \text{(D-8)}$$

$$.5\left(\frac{\partial^2 f}{\partial x_3^2}\right)_{T,P,x_{j\neq 3}} = C6 \qquad \text{(D-9)}$$

$$\left(\frac{\partial^2 f}{\partial x_1 \partial x_2}\right)_{T,P,x_{j\neq i}} = C7 \qquad (D\text{-}10)$$

$$\left(\frac{\partial^2 f}{\partial x_1 \partial x_3}\right)_{T,P,x_{j\neq i}} = C8 \qquad (D\text{-}11)$$

$$\left(\frac{\partial^2 f}{\partial x_2 \partial x_3}\right)_{T,P,x_{j\neq i}} = C8 \qquad (D\text{-}12)$$

where f represents equation (D-2) and C1 through C8 represent the derivatives evaluated at the base point $(x_1^\circ, x_2^\circ, x_3^\circ)$ around which the Taylor series was expanded. A Taylor series expansion of equation (D-2) around these three base points can now be written as:

$$RT\ln\gamma_1 - RT\ln\gamma_1^0 = C1(x_1-x_1^\circ) + C2(x_2-x_2^\circ) + C3(x_3-x_3^\circ) + C4(x_1-x_1^\circ)^2 + C5(x_2-x_2^\circ)^2 + C6(x_3-x_3^\circ)^2 + C7(x_1-x_1^\circ)(x_2-x_2^\circ) + C8(x_1-x_1^\circ)(x_3-x_3^\circ) + C9(x_2-x_2^\circ)(x_3-x_3^\circ) \quad \text{(D-13)}$$

If we now define σ° as:

$$\sigma^\circ = x_1^\circ + x_2^\circ + x_3^\circ \quad \text{(D-14)}$$

and assume that any one of the following three assumptions holds:

(1) $\sigma \to 1$ and $\sigma^\circ \to 1$ as we approach convergence

(2) $\sigma \to \sigma^\circ$ ($\neq 1$) as we approach convergence

(3) $\sigma - \sigma^\circ$ becomes small relative to the other terms as we approach convergence,

we can substitute for x_1° and x_1 in equation (D-13) to obtain:

$$RT\ln\gamma_1 - RT\ln\gamma_1^0 = (-C1 + C2)(x_2-x_2^\circ) + (-C1 + C3)(x_3-x_3^\circ) + (C4 + C5 - C7)(x_2-x_2^\circ)^2 + (C4 + C6 - C8)(x_3-x_3^\circ)^2 + (2C4 - C7 - C8 + C9)(x_2-x_2^\circ)(x_3-x_3^\circ) \quad \text{(D-15)}$$

If the constants in equation (D-15) are combined, we get:

$$RT\ln\gamma_1 - RT\ln\gamma_1^0 = (2Ax_2^\circ/(\sigma^\circ)^2 + Cx_3^\circ/(\sigma^\circ)^2)(x_2-x_2^\circ) + (2Bx_3^\circ/(\sigma^\circ)^2 + Cx_2^\circ/(\sigma^\circ)^2)(x_3-x_3^\circ) + (A/(\sigma^\circ)^2)(x_2-x_2^\circ)^2 + (B/(\sigma^\circ)^2)(x_3-x_3^\circ)^2 + (C/(\sigma^\circ)^2)(x_2-x_2^\circ)(x_3-x_3^\circ) \quad \text{(D-16)}$$

However, any one of the three assumptions considered will further simplify equation (D-16) to:

$$RT\ln\gamma_1 - RT\ln\gamma_1^0 = (2A(x_2^*)^\circ + C(x_3^*)^\circ)(x_2^*-(x_2^*)^\circ) + (2B(x_3^*)^\circ + C(x_2^*)^\circ)(x_3^*-(x_3^*)^\circ) + (A)(x_2^*-(x_2^*)^\circ)^2 + (B)(x_3^*-(x_3^*)^\circ)^2 + (C)(x_2^*-(x_2^*)^\circ)(x_3^*-(x_3^*)^\circ) \quad \text{(D-17)}$$

where the mole numbers have been converted back to mole fractions.

The pseudo-quartenary two-suffix Margules equation may

be written as:

$$RT\ln\gamma_1 = A(x_2^*)^2 + B(x_3^*)^2 + C(x_4^*)^2 + Dx_2^*x_3^* + Ex_2^*x_4^* + Fx_3^*x_4^* \quad \text{(D-18)}$$

where, in addition to the terms in equation (D-1), a fourth component is now included. Using the same procedure described above, the following expansion is obtained:

$$\begin{aligned} RT\ln\gamma_1 - RT\ln\gamma_1^o = {} & (2A(x_2^*) + D(x_3^*) + E(x_4^*))(x_2^* - (x_2^*)^\circ) \\ & + (2B(x_3^*) + D(x_2^*) + F(x_4^*))(x_3^* - (x_3^*)^\circ) \\ & + (A)(x_2^* - (x_2^*)^\circ)^2 \\ & + (B)(x_3^* - (x_3^*)^\circ)^2 \\ & + (C)(x_4^* - (x_4^*)^\circ)^2 \\ & + (D)(x_2^* - (x_2^*)^\circ)(x_3^* - (x_3^*)^\circ) \\ & + (E)(x_2^* - (x_2^*)^\circ)(x_4^* - (x_4^*)^\circ) \\ & + (F)(x_3^* - (x_3^*)^\circ)(x_4^* - (x_4^*)^\circ) \end{aligned} \quad \text{(D-19)}$$

The bracketed terms in equations (D-17) and (D-19) that contain only constants are replaced by C1-C5 and C1-C9 , respectively, in equations (40) and (41).

Appendix E

Mutual Solubility Calculations

In the solvent selection scheme, a hypothetical binary mixture was created for each binary pair i-j in an n component system. Mutual solubiltiy estimates can be calculated for each of these pairs. The two phase mixture may be described by the following mole fraction combinations:

$$x_i^R + x_j^R = 1 \qquad \text{(E-1)}$$

$$x_j^E + x_i^E = 1 \qquad \text{(E-2)}$$

where R and E denote a raffinate (i-rich phase) and an extract (j-rich phase) phase respectively, and $x_i^R = x_j^E = .98$ and $x_j^R = x_i^E = .02$. K-values may be represented by:

$$K_i = \frac{x_i^E}{x_i^R} \qquad \text{(E-3)}$$

$$K_j = \frac{x_j^E}{x_j^R} \qquad \text{(E-4)}$$

Using equations (E:1-4), the following relationships are developed:

$$x_i^E = 1 - x_j^E = x_i^R K_i \qquad \text{(E-5)}$$

$$x_j^E = x_j^R K_j \qquad \text{(E-6)}$$

$$x_j^E = (1 - x_i^R) K_j \qquad \text{(E-7)}$$

Combining equations (E-5) and (E-7) gives:

$$x_i^R = (1-K_i)/(K_i-K_j) \qquad \text{(E-8)}$$

From equations (E-8),(E-3), and (E-2) the mutual solubilities can be expressed as:

$$x_i^E = K_i \left[\frac{1-K_j}{K_i-K_j}\right] \qquad \text{(E-9)}$$

$$x_j^R = 1 - \left[\frac{1-K_j}{K_i-K_j}\right] \qquad \text{(E-10)}$$

where x_i^E is defined as the solubility of component i in the extract phase and x_j^R is defined as the solubility of

component j in the raffinate phase. (N.B. Choosing which phase is the raffinate and which phase is the extract is done arbitrarily. If the mixture definition was set up in the reverse manner, the derived expressions would give the same values as equations (E-9) and (E-10).)

Appendix F

Criterion for Total Miscibility

Before mutual solubilities are calculated in the solvent selection scheme, we must make sure that the mixture displays partial miscibility. When A and B denote the two solvents and K_A and K_B are their K-values, the following constitutive, mass balance and equilibrium equations can be presented:

$$\sum_{i}^{2} y_i - x_i = 0 \qquad \text{(F-1)}$$

$$\alpha y_A + (1-\alpha)x_A = z_A \qquad \text{(F-2)}$$

$$\alpha y_B + (1-\alpha)x_B = z_B \qquad \text{(F-3)}$$

$$y_i = K_i x_i \qquad \text{(F-4)}$$

where x and y are the more dense and less dense phase fractions respectively, i denotes either A or B, and α is the phase split fraction. When equation (F-4) is substituted into equation (F-1) we get:

$$\sum_{i}^{2} x_i(K_i-1) \qquad \text{(F-5)}$$

Equation (F-4) is then substituted into equations (F-2) and (F-3). The results may be represented in terms of X_A and X_B as:

$$x_A = \frac{z_A}{1+\alpha(K_A-1)} \qquad \text{(F-6)}$$

and:

$$x_B = \frac{z_B}{1+\alpha(K_B-1)} \qquad \text{(F-7)}$$

Equations (F-6) and (F-7) are substituted into equation (F-5). When the summation is evaluated:

$$\frac{z_A(K_A-1)}{1+\alpha(K_A-1)} + \frac{z_B(K_B-1)}{1+\alpha(K_B-1)} = 0 \qquad \text{(F-8)}$$

If z_A and z_B are equated to .5 and substituted into equation (F-8), the result is:

$$\frac{K_A-1}{1+\alpha(K_A-1)} + \frac{K_B-1}{1+\alpha(K_B-1)} = 0 \qquad \text{(F-9)}$$

Let $\beta = 1/(K_i-1)$. Equation (F-9) becomes:

$$\frac{1}{\beta_A+\alpha} + \frac{1}{\beta_B+\alpha} \qquad \text{(F-10)}$$

Solving for α:

$$\alpha = -.5(\beta_A+\beta_B) \qquad \text{(F-11)}$$

For immiscibility to occur, α has to have a value between zero and one:

$$-2 < \beta_A + \beta_B < 0 \qquad \text{(F-12)}$$

or:

$$-2 < \left[\frac{1}{K_A-1} + \frac{1}{K_B-1} \right] < 0 \qquad \text{(F-13)}$$

Appendix G

Composition Derivatives of Pseudo-quartenary Two-suffix Margules Equation

The pseudo-quartenary two-suffix Margules equation may be written as:

$$RT\ln\gamma_i = A_i(x_j^*)^2+B_i(x_k^*)^2+C_i(x_m^*)^2+D_i x_j^* x_k^*+E_i x_j^* x_m^*+F_i x_k^* x_m^* \quad (G\text{-}1)$$

where A_i, Bi, C_i, D_i, E_i and F_i are constant coefficients for component i and x_j^*, x_k^*, and x_m^* are pseudo-component mole fractions. The procedure for obtaining the composition derivatives is the same as the one described for the ternary model. The following derivatives are obtained:

$$\frac{\partial\gamma_i}{\partial x_i} = -2(\gamma_i)\ln\gamma_i \quad (G\text{-}2)$$

$$\frac{\partial\gamma_i}{\partial x_j} = -2(\gamma_i)\ln\gamma_i + 2A_i(x_j^*) + D_i(x_k^*) + E_i(x_m^*) \quad (G\text{-}3)$$

$$\frac{\partial \gamma_i}{\partial x_k} = -2(\gamma_i)\ln\gamma_i + 2B_i(x_k^*) + D_i(x_j^*) + F_i(x_m^*) \quad \text{(G-4)}$$

$$\frac{\partial \gamma_i}{\partial x_m} = -2(\gamma_i)\ln\gamma_i + 2C_i(x_m^*) + E_i(x_j^*) + F_i(x_k^*) \quad \text{(G-5)}$$

where x_j^*, x_k^* and x_m^* are mole fractions.

Appendix H

Composition Derivatives of Margules Expansions

The pseudo-ternary two-suffix Margules Taylor series expansion is:

$$\begin{aligned} RT\ln\gamma_i = {} & RT\ln\gamma_i^{\circ} + C1(x_B^*-(x_B^*)^{\circ}) + C2(x_C^*-(x_C^*)^{\circ}) \\ & + C3(x_B^*-(x_B^*)^{\circ})^2 + C4(x_C^*-(x_C^*)^{\circ})^2 \\ & + C5(x_B^*-(x_B^*)^{\circ})(x_C^*-(x_C^*)^{\circ}) \end{aligned} \qquad \text{(H-1)}$$

The partial derivatives of γ_i with respect to the mole numbers of the component of interest , i, pseudo-component B, and pseudo-component C are:

$$\frac{\partial\gamma_i}{\partial x_i} = \frac{\gamma_i}{RT}\left[\begin{array}{l} -C1(x_B^*)-C2(x_C^*)-2C3(x_B^*)(x_B^*-(x_B^*)^{\circ}) \\ -2C4(x_C^*)(x_C^*-(x_C^*)^{\circ})-C5(x_B^*)(x_C^*-(x_C^*)^{\circ}) \\ -C5(x_C^*)(x_B^*-(x_B^*)^{\circ}) \end{array}\right] \qquad \text{(H-2)}$$

$$\frac{\partial\gamma_i}{\partial x_B} = \frac{\gamma_i}{RT}\left[\begin{array}{l} C1(1-x_B^*)-C2(x_C^*)+2C3(1-x_B^*)(x_B^*-(x_B^*)^{\circ}) \\ -2C4(x_C^*)(x_C^*-(x_C^*)^{\circ})+C5(1-x_B^*)(x_C^*-(x_C^*)^{\circ}) \\ -C5(x_C^*)(x_B^*-(x_B^*)^{\circ}) \end{array}\right] \qquad \text{(H-3)}$$

$$\frac{\partial \gamma_i}{\partial x_C} = \frac{\gamma_i}{RT}\left(\begin{array}{l} -C1(x_B^*)+C2(1-x_C^*)-2C3(x_B^*)(x_B^*-(x_B^*)^\circ) \\ +2C4(1-x_C^*)(x_C^*-(x_C^*)^\circ)-C5(x_B^*)(x_C^*-(x_C^*)^\circ) \\ +C5(1-x_C^*)(x_B^*-(x_B^*)^\circ) \end{array}\right) \qquad \text{(H-4)}$$

The Taylor series expansion of the two-suffix pseudo-quartenary Margules equation is:

$$\begin{aligned} RT\ln\gamma_i = {} & RT\ln\gamma_i^\circ + C1(x_B^*-(x_B^*)^\circ) + C2(x_C^*-(x_C^*)^\circ) \\ & + C3(x_D^*-(x_D^*)^\circ) + C4(x_B^*-(x_B^*)^\circ)^2 \\ & + C5(x_C^*-(x_C^*)^\circ)^2 + C6(x_D^*-(x_D^*)^\circ)^2 \\ & + C7(x_B^*-(x_B^*)^\circ)(x_C^*-(x_C^*)^\circ) \\ & + C8(x_B^*-(x_B^*)^\circ)(x_D^*-(x_D^*)^\circ) \\ & + C9(x_C^*-(x_C^*)^\circ)(x_D^*-(x_D^*)^\circ) \end{aligned} \qquad \text{(H-5)}$$

The composition derivatives are:

$$\frac{\partial \gamma_i}{\partial x_i} = \left(\begin{array}{l} -C1(x_B^*)-C2(x_C^*)-C3(x_D^*) \\ -2C4(x_B^*)(x_B^*-(x_B^*)^\circ)-2C5(x_C^*)(x_C^*-(x_C^*)^\circ) \\ -2C6(x_D^*)(x_D^*-(x_D^*)^\circ) \\ -C7(x_B^*)(x_C^*-(x_C^*)^\circ)-C7(x_C^*)(x_B^*-(x_B^*)^\circ) \\ -C8(x_B^*)(x_D^*-(x_D^*)^\circ)-C8(x_D^*)(x_B^*-(x_B^*)^\circ) \\ -C9(x_C^*)(x_D^*-(x_D^*)^\circ)-C9(x_D^*)(x_C^*-(x_C^*)^\circ) \end{array}\right) \qquad \text{(H-6)}$$

$$\frac{\partial \gamma_i}{\partial x_B} = \begin{pmatrix} C1(1-x_B^*)-C2(x_C^*)-C3(x_D^*) \\ +2C4(1-x_B^*)(x_B^*-(x_B^*)^\circ)-2C5(x_C^*)(x_C^*-(x_C^*)^\circ) \\ -2C6(x_D^*)(x_D^*-(x_D^*)^\circ) \\ +C7(1-x_B^*)(x_C^*-(x_C^*)^\circ)-C7(x_C^*)(x_B^*-(x_B^*)^\circ) \\ +C8(1-x_B^*)(x_D^*-(x_D^*)^\circ)-C8(x_D^*)(x_B^*-(x_B^*)^\circ) \\ -C9(x_C^*)(x_D^*-(x_D^*)^\circ)-C9(x_D^*)(x_C^*-(x_C^*)^\circ) \end{pmatrix} \quad \text{(H-7)}$$

$$\frac{\partial \gamma_i}{\partial x_C} = \begin{pmatrix} -C1(x_B^*)+C2(1-x_C^*)-C3(x_D^*) \\ -2C4(x_B^*)(x_B^*-(x_B^*)^\circ)+2C5(1-x_C^*)(x_C^*-(x_C^*)^\circ) \\ -2C6(x_D^*)(x_D^*-(x_D^*)^\circ) \\ -C7(x_B^*)(x_C^*-(x_C^*)^\circ)+C7(1-x_C^*)(x_B^*-(x_B^*)^\circ) \\ -C8(x_B^*)(x_D^*-(x_D^*)^\circ)-C8(x_D^*)(x_B^*-(x_B^*)^\circ) \\ +C9(1-x_C^*)(x_D^*-(x_D^*)^\circ)-C9(x_D^*)(x_C^*-(x_C^*)^\circ) \end{pmatrix} \quad \text{(H-8)}$$

$$\frac{\partial \gamma_i}{\partial x_D} = \begin{pmatrix} -C1(x_B^*)-C2(x_C^*)+C3(1-x_D^*) \\ -2C4(x_B^*)(x_B^*-(x_B^*)^\circ)-2C5(x_C^*)(x_C^*-(x_C^*)^\circ) \\ +2C6(1-x_D^*)(x_D^*-(x_D^*)^\circ) \\ -C7(x_B^*)(x_C^*-(x_C^*)^\circ)-C7(x_C^*)(x_B^*-(x_B^*)^\circ) \\ -C8(x_B^*)(x_D^*-(x_D^*)^\circ)+C8(1-x_D^*)(x_B^*-(x_B^*)^\circ) \\ -C9(x_C^*)(x_D^*-(x_D^*)^\circ)+C9(1-x_D^*)(x_C^*-(x_C^*)^\circ) \end{pmatrix} \quad \text{(H-9)}$$

Appendix I

The Jacobian Matrix

The multivariable Newton-Raphson convergence algorithm requires the partial derivatives of the defining equations with respect to all the independent variables. The independent variables in this work are the phase split fraction, α and the phase compositions, x and y (for an n-component system there are n of each phase composition and one α for a total of 2n+1 variables). The 2n+1 defining equations are:

$$f_i = z_i - \alpha_i y_i - (1-\alpha_i)x_i \qquad \text{(I-1)}$$

$$g_i = \gamma_i^x x_i - \gamma_i^y y_i \qquad \text{(I-2)}$$

$$h = \sum (y_i - x_i) \qquad \text{(I-3)}$$

where i represents each of the components in a mixture. The following partial derivatives are easily obtained:

$$\partial f_i / \partial x_i = \alpha - 1 \qquad \text{(I-4)}$$

$$\partial f_i/\partial y_i = -\alpha \qquad \text{(I-5)}$$

$$\partial f_i/\partial \alpha = x_i - y_i \qquad \text{(I-6)}$$

$$\partial g_i/\partial x_i = \gamma_i^x + x_i(\partial \gamma_i^x/\partial x_i) \qquad \text{(I-7)}$$

$$\partial g_i/\partial y_i = -\gamma_i - y_i(\partial \gamma i/\partial y_i) \qquad \text{(I-8)}$$

$$\partial g_i/\partial \alpha = 0.0 \qquad \text{(I-9)}$$

$$\partial h/\partial x_i = -1.0 \qquad \text{(I-10)}$$

$$\partial h/\partial y_i = 1.0 \qquad \text{(I-11)}$$

$$\partial h/\partial \alpha = 0.0 \qquad \text{(I-12)}$$

where each differential considers all 'other' variables as constant. For a three component example system (n = 3), the derivative matrix (Jacobian) has the following arrangement:

	$\partial/\partial x_1$	$\partial/\partial x_2$	$\partial/\partial x_3$	$\partial/\partial y_1$	$\partial/\partial y_2$	$\partial/\partial y_3$	$\partial/\partial \alpha$
f_1							
f_2							
f_3							
g_1							
g_2							
g_3							
h							

where each location in this 2n+1 x 2n+1 matrixx (in this example, 7x7) is the result of the column differential operating on the row function.

Appendix J

Geometric Regression Point Selection Scheme

This work considers the base point to be the most recent values of x_i or y_i at the time the LA model was created. Additional composition sets are found from this base point by geometric perturbation (Bryan, 1985). The following steps are taken.

1) Reduce the base point to a pseudo-ternary point by grouping the compositions according to the following scheme (N.B. the pseudo-compositions defined here are unique and are not the same as the ones used in the various models).

    a) All the solute component compositions are combined into a pseudo-composition xA.

    b) The composition of solvent IV is placed in xB.

    c) The composition of solvent IL is placed in xC.

2) The base point can now be placed on a ternary diagram in which the left and right vertices are 100% xB and 100% xC, respectively. We now want to obtain additional points within the proximity of the base point. They can not be haphazerdly placed, as the two-phase region should not contain any points. Since the base point is close to the binodal curve (it is assumed that we are dealing with a type I system. Type II systems are less sensitive to placement of points in two-phase areas and will usually allow this scheme to be used (Bryan, 1985)), it would be logical to keep all other points no closer to the phase envelope than this base point. When the majority solvent of the phase of interest is IV, the base point will be placed closer to 100% xB than 100% xC. If we locate the portion of the binodal curve nearest to the base point and draw a tangent line, the line will have a positive slope. The same argument for the majority solvent IL will result in a tangent line of negative slope. The orientation of the base point and its perturbations can be set up so it approximates the orientation of the tangent lines.

3) The procedure for achieving this orientation is:

a) Imagine a line 1 drawn from the base point to the 100% composition vertex of the minority solvent. Imagine a second line 2 perpendicular to the first line. Then:

$$m_1 = -1/m_2 \qquad \text{(J-1)}$$

where m_i is the slope of line i. The angle line 2 makes with the horizontal, α, is related to m_2 by:

$$m_2 = \tan(\alpha) \qquad \text{(J-2)}$$

b) Consider 100% xB to be the origin of a two dimensional rectangular coordinate system. Select i sets of points x_i', y_i' on a semi-circle of radius r with center 0.0,0.0.

c) Assume the length of a side of the ternary diagram (equilateral triangle) is 1.0. Transform the base point to rectangular

coordinates with 100% xB as the origin. The coordinates of this base point,x0 and y0 are:

$$x0 = 1 - xB - xA/2 \qquad (J-3)$$

$$y0 = (xA).866 \qquad (J-4)$$

d) The slope of Line 1 described in (a) can now be expressed in terms of x0 and y0:

a_1) If IV is the majority solvent:

$$m_1 = -y0/(1-x0) \qquad (J-5)$$

a_2) If IL is the majority solvent:

$$m_1 = y0/x0 \qquad (J-6)$$

e) Equation (J-2) can be expressed as:

$$\alpha = \arctan(-1/m_1) \qquad (J-7)$$

where m_1 will be represented by either

equation (J-5) or (J-6).

f) The semi-circle of points set up in (b) is now translated to the new origin x0,y0. It is then rotated by α. The new set of points x_i, y_i in terms of the origin at 100% xB is :

$$x_i = x0 + (\acute{x_i})\cos(\alpha) - (\acute{y_i})\sin(\alpha) \qquad \text{(J-8)}$$

$$y_i = y0 + (\acute{x_i})\sin(\alpha) + (\acute{y_i})\cos(\alpha) \qquad \text{(J-9)}$$

4) The sets of x_i, y_i are now converted back to triangular coordinates by using equations (J-3) and (j-4) :

$$xA_i = y_i/.866 \qquad \text{(J-10)}$$

$$xB_i = 1 - .5(xA_i) - x_i \qquad \text{(J-11)}$$

$$xC_i = 1 - xA_i - xB_i \qquad \text{(J-12)}$$

5) These are converted back to real compositions :

$$xreal(IV)_i = xB_i \quad \text{(J-13)}$$

$$xreal(IL)_i = xC_i \quad \text{(J-14)}$$

$$xreal_{i \neq IV,IL} = (xA_i)x_{i \neq IV,IL} \; / \; \Sigma \; x_{i \neq IV,IL} \quad \text{(J-15)}$$

where the solute pseudo-components have been converted back to real compositions according to the proportion of each solute in the solute mixture.

Appendix K

Comparison of γ_i and $\partial\gamma_i/\partial x_j$ Provided by LA Models

Activity coefficients and their composition derivatives used by some of the LA methods are compared to MT values. The example (LLE) system used for these comparisons is heptane-DMF-water-Benzene at 293.15 K.

Table K-1. Comparison of γ_i Used by LAs With MT Values (Liquid Phase)
(System 8, example # 1)

It. #	Cmpt. #	Activity Coefficient Model LA1	LA2	LA5(b)	LA6(b)	NRTL
	i			γ_i		
First Iteration	1	150	142	146	145	146
	2	.647	.643	.650	.648	.650
	3	.947	.945	.948	.951	.947
	4	8.06	8.03	7.89	7.88	7.89
Middle Iteration	1	142	144	144	142	142
	2	.646	.648	.648	.646	.646
	3	.954	.951	.951	.955	.954
	4	7.73	7.83	7.83	7.73	7.73
Final Iteration	1	141	142	142	141	141
	2	.646	.646	.646	.646	.646
	3	.955	.955	.955	.955	.955
	4	7.72	7.72	7.72	7.72	7.72

Table K-2. Comparison of γ_i Used by LAS With MT Values (Vapor Phase)
(System 8, example # 1)

It.	Cmpt.	LA1	LA2	LA5(b)	LA6(b)	NRTL
	i			γ_i		
First Iteration	1	1.32	1.32	1.32	1.32	1.32
	2	2.90	2.87	2.90	2.90	2.91
	3	101	97.6	97.6	97.1	98.4
	4	1.07	1.07	1.07	1.08	1.07
Middle Iteration	1	1.32	1.32	1.32	1.31	1.31
	2	2.92	2.91	2.91	2.92	2.92
	3	99.8	97.9	97.9	99.9	99.8
	4	1.08	1.07	1.07	1.08	1.08
Final Iteration	1	1.31	1.31	1.31	1.31	1.31
	2	2.93	2.92	2.92	2.93	2.93
	3	100	99.9	99.9	100	100
	4	1.08	1.08	1.08	1.08	1.08

Table K-3. Comparison of Activity Coefficient Derivatives Used By LA6(b) and LA1 With MT Values (System 8, example 1)

Middle Iteration (LA6[b])

LA6(b) $\partial\gamma_i/\partial x_j$

i \ j	1	2	3	4
1	-2385.6	-284.9	405.6	-1187.7
2	-1.39	0.80	-0.46	-1.53
3	2.68	-0.67	0.21	2.55
4	-72.4	-18.4	21.2	-42.2

NRTL $\partial\gamma_i/\partial x_j$ At LA6(b) Liquid Phase Compositions

i \ j	1	2	3	4
1	-2235.7	-280.0	398.7	-1175.7
2	-1.28	0.79	-0.46	-1.50
3	2.69	-0.68	0.22	2.55
4	-64.2	-17.9	20.7	-41.6

Middle Iteration (LA1)

LA1 $\partial\gamma_i/\partial x_j$

i \ j	1	2	3	4
1	- 1447.9	-556.5	543.5	-556.53
2	-1.49	0.57	-0.27	-1.49
3	2.61	-0.53	0.09	2.61
4	-31.7	2.38	2.38	-32.7

NRTL $\partial\gamma_i/\partial x_j$ At LA1 Liquid Phase Compositions

i \ j	1	2	3	4
1	-2242.3	-281.1	399.6	-1179.3
2	-1.28	0.79	-0.46	-1.50
3	2.69	-0.67	0.22	2.55
4	-64.32	-18.0	20.7	-41.7

Appendix L

Listing of Computer Code

```
c                Main Program

c        The various schemes that will perform
c        single stage (constant T & P) flash
c        calculations are set up.
c        ****************************************
c
         IMPLICIT REAL*8 (A-H,O-Z)
         COMMON/OPT/NSS,KIND,NMT,NLA1
         COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
         COMMON/SHOT/IFILE
         COMMON/INIT/XLIM(10,10),IV,IL
         COMMON/ANTOINE/ANTA(20),ANTB(20),ANTC(20)
         COMMON/PVAP/PSTAR(20)
         COMMON/RPOINTS/IDP
         REAL*8 DUM(21,21),DUMINV(21,21)
         INTEGER HALT
c        OPEN(FILE='INPUTT',UNIT=10,STATUS='UNKNOWN')
c        OPEN(FILE='FEED',UNIT=20,STATUS='UNKNOWN')
c        ****************************************
c        Specify the type of system that will be
c        flashed;liquid-liquid or vapor-liquid.

5        WRITE(6,*)
10       WRITE(6,15)
15       FORMAT(/,10X,'L-L-E (TYPE 1) OR V-L-E (TYPE 2) ?')
         READ(6,*)KIND
c        ****************************************
c        The number of components in the flash
c        mixture is set up.

         WRITE(6,20)
20       FORMAT(/,10X,'INPUT NUMBER OF COMPONENTS')
         READ(6,*)N
c        ****************************************
c        The binary parameters of the NRTL
c        equation for the system of interest
c        are obtained.

         IF(KIND.EQ.1)CALL LLDATA
         IF(KIND.EQ.2)CALL VLDATA
c        ****************************************
c        All flash calculations will occur at
c        atmospheric pressure (mmHg).

         P=760.
c        ****************************************
c        The flash temperature is set
```

```
c       (degrees Kelvin).

        WRITE(6,25)
25      FORMAT(/,10X,'INPUT TEMPERATURE(K)')
        READ(6,*)T
c       ****************************************
c       The vapor pressures of the VLE mixture
c       components are obtained.

        IF(KIND.EQ.2)CALL PVAPOR
c       ****************************************
c       The mixture feed fractions are set up.

        WRITE(6,30)
30      FORMAT(/,10X,'INPUT Z(I)S')
        READ(6,*)(Z(I),I=1,N)
c       ****************************************
c       The various inputs can be obtained from
c       data files 10 and 20.

c       READ(10,*)KIND,N,IFILE,T,NSS,NMT,NLA1,HALT
c       READ(20,*)(Z(I),I=1,N)
c       ****************************************
c       Initialize process variables. The solvent
c       components in LLE systems and the key
c       components in VLE systems are determined.

        IF(KIND.NE.1)GOTO 35
                CALL SOLVNT
                CALL INITIL
35      IF(KIND.NE.2)GOTO 40
                CALL KEY
                CALL INITIV
c       ****************************************
c       The iterative scheme or combination of
c       iterative schemes is set up. NSS = # of
c       successive substitution iterations to be
c       performed using activity coefficients provided
c       by the NRTL equation. NMT = # of multivariable
c       Newton-Raphson iterations to be performed
c       using activity coefficients and their composition
c       derivatives from the NRTL equation. NLA1 =
c       same as NMT except activity coefficients
c       and their derivatives are obtained from a
c       simplified expression (pseudo-ternary,
c       pseudo-quartenary,pseudo-ternary expansion,
c       or pseudo-quartenary expansion forms of
c       the two-suffix Margules equation). NLA2
c       = same as NMT except activity coefficients
```

```
c       and their derivatives are provided by a
c       'true' linearized expansion of the two-suffix
c       Margules equation (LTS model).

40      WRITE(6,45)
45      FORMAT(/,10X,'INPUT # OF SS,MR,AND LA ITERATIONS')
        READ(6,*)NSS,NMT,NLA1,NLA2
c       ****************************************
c       Specify the size of the matrix of material
c       balance equations that will be used in the
c       multivariable NR computational scheme.

        ISIZE=(2*N)+1
c       ****************************************
c       If the NLA1 method is to be implemented,one
c       of the following forms of the two-suffix
c       Margules equation is specified. These may
c       be called local approximations(LAs).

c       1:pseudo-ternary 2-suffix Margules
c       2:pseudo-quartenary 2-suffix Margules
c       3:expanded pseudo-ternary
c       4:expanded pseudo-quartenary

        IF(NLA1.EQ.0)GOTO 50
        WRITE(6,*)'INPUT MODEL'
        READ(6,*)MODEL
c       ****************************************
c       Number of LA model parameters is specified

        IF(MODEL.EQ.1)NUM=3
        IF(MODEL.EQ.2)NUM=6
        IF(MODEL.EQ.3)NUM=5
        IF(MODEL.EQ.4)NUM=9
c       ****************************************
c       The successive substitution (SSUBST)
c       algorithm is run to convergence or
c       is used to provide the starting point
c       for the multivariable Newton-Raphson
c       algorithms. The latter will use activity
c       coefficients provided by a complex
c       molecular thermodynamic model (NRMT)
c       or a simple analytic model (NRLA1 or NRLA2).

50      EPS=5.E-05
        IF(NSS.NE.0)CALL SSUBST(NSS,EPS,KIND,NLA2)
        IF(NMT.NE.0)CALL NRMT(NMT,ISIZE,DUM,DUMINV,KIND)
        IF(NLA1.NE.0)CALL NRLA1(NLA1,MODEL,ISIZE,DUM,
     &  DUMINV,KIND,NUM)
```

```
      IF(NLA2.NE.0)CALL NRLA2(NLA2,ISIZE,DUM,DUMINV,KIND)
c     ***************************************
c     The user may now terminate the session or
c     perform another flash.

      WRITE(6,55)
55    FORMAT(/,10X,'EXIT?1-YES,2-NO')
      READ(6,*)HALT
      IF(HALT.EQ.2)GOTO 5
      STOP
      END
```

```
                SUBROUTINE  VLDATA

c       Vapor pressures and NRTL parameters are
c       obtained for VLE system components.
c       ***************************************
        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/CHARACTER/NAME(20)
        COMMON/SHOT/IFILE
        COMMON/ANTOINE/ANTA(20),ANTB(20),ANTC(20)
        OPEN(FILE='AMW',UNIT=25,STATUS='UNKNOWN')
        OPEN(FILE='MEAW',UNIT=30,STATUS='UNKNOWN')
        OPEN(FILE='HBE',UNIT=35,STATUS='UNKNOWN')
        OPEN(FILE='CBTE',UNIT=40,STATUS='UNKNOWN')
        OPEN(FILE='HHCBTE',UNIT=45,STATUS='UNKNOWN')
c       ***************************************
        WRITE(6,10)
10      FORMAT(/,10X,'INPUT DATA FILE')
        READ(6,*)IFILE
c       ***************************************
c       Names of system components are read in
c       from top line of system data file

        NN=2*N
        READ(IFILE,20)(NAME(I),I=1,NN)
20      FORMAT(1X,20A4)
c       ***************************************
c       Antoine constants and Cij,Cji,and alpha
c       are read in from system data file

        DO 30 I=1,N
                ALPHA(I,I)=1.
                C(I,I)=0.
30      CONTINUE
        NM=N-1
        DO 50 J=1,NM
               JJ=J+1
                DO 40 I=JJ,N
                    READ(IFILE,*)C(J,I),C(I,J),ALPHA(J,I)
                    ALPHA(I,J)=ALPHA(J,I)
40              CONTINUE
50      CONTINUE
        DO 60 I=1,N
                READ(IFILE,*)ANTA(I),ANTB(I),ANTC(I)
60      CONTINUE
        REWIND(IFILE)
        RETURN
        END
                 SUBROUTINE PVAPOR
```

```
c       Vapor pressures of VLE components are
c       calculated using the Antoine equation.
c       ***************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/PVAP/PSTAR(20)
        COMMON/ANTOINE/ANTA(20),ANTB(20),ANTC(20)
c       ***************************************
        DO 10 I=1,N
                PSTAR(I)=0.
                PSTAR(I)=DEXP(ANTA(I)-ANTB(I)/(T+ANTC(I)))
10      CONTINUE
        RETURN
        END
```

```
                SUBROUTINE LLDATA

c       Vapor pressures and NRTL parameters are
c       given for LLE system components.
c       ****************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/CHARACTER/NAME(20)
        COMMON/SHOT/IFILE
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        OPEN(FILE='BTCHHS',UNIT=50,STATUS='UNKNOWN')
        OPEN(FILE='DWB',UNIT=55,STATUS='UNKNOWN')
        OPEN(FILE='HDEB',UNIT=60,STATUS='UNKNOWN')
        OPEN(FILE='ICBF',UNIT=65,STATUS='UNKNOWN')
        OPEN(FILE='AMO',UNIT=70,STATUS='UNKNOWN')
c       ****************************************
        WRITE(6,10)
10      FORMAT(/,10X,'INPUT DATA FILE' )
        READ(6,*)IFILE
c       ****************************************
c       Names of system components are read in
c       from top line of system data file

        NN=2*N
        READ(IFILE,20)(NAME(I),I=1,NN)
20      FORMAT(1X,20A4)
c       ****************************************
c       Antoine constants and Cij,Cji,and alpha
c       are read in from system data file

        DO 30 I=1,N
                ALPHA(I,I)=1.
                C(I,I)=0.
30      CONTINUE
        NM=N-1
        DO 50 J=1,NM
               JJ=J+1
                DO 40 I=JJ,N
                    READ(IFILE,*)C(J,I),C(I,J),ALPHA(J,I)
                    ALPHA(I,J)=ALPHA(J,I)
40              CONTINUE
50      CONTINUE
        REWIND(IFILE)
        RETURN
        END
```

```
                SUBROUTINE SOLVNT

c       Pair of solvents is chosen by selecting
c       the least miscible binary pair. This is
c       used with LLE systems.
c       ****************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/INIT/XLIM(10,10),IV,IL
        REAL*8 KA,KB,KMAX,KMIN
c       ****************************************
        KMAX=1
        DO 20 I=1,10
                DO 10 J=1,10
                        XLIM(I,J)=1
10              CONTINUE
20      CONTINUE
c       ****************************************
c       Solvent IV (the 'light' phase solvent) is
c       selected.

        N1=N-1
        IV=0
        DO 60 IA=1,N1
           I1=IA+1
              DO 50 IB=I1,N
                 CALL BINGAM(IA,IB,GLA,GLB)
                 CALL BINGAM(IB,IA,GVB,GVA)
                 KA=GLA/GVA
                 KB=GLB/GVB
                 A=-.5*((1./(KA-1.))+(1./(KB-1)))
                 IF(A.LE.0..OR.A.GE.1.)GOTO 30
                 XA=(KB-1.)/(KB-KA)
                 XLIM(IA,IB)=KA*XA
                 XLIM(IB,IA)=1.-XA
30               IF(KB.LT.KMAX.OR.Z(IB).LT.0.1)GOTO 40
                 KMAX=KB
                 IV=IB
40               IF((1./KA).LT.KMAX.OR.Z(IA).LT.0.1)GOTO 50
                 KMAX=1./KA
                 IV=IA
50            CONTINUE
60      CONTINUE

        IF(IV.NE.0)GOTO 80

        WRITE(6,70)
70      FORMAT(/,10X,'NO V-PHASE SOLVENT FOUND. INPUT IV')
```

```
        READ(6,*)IV
c       ****************************************
c       Solvent IL (the 'heavy' phase solvent) is
c       selected.

80      KMIN=1
        IL=0

        DO 90 I=1,N
                IF(I.EQ.IV)GOTO 90
                CALL BINGAM(I,IV,GLA,GLB)
                CALL BINGAM(I,IV,GVB,GVA)
                KA=GLA/GVA
                IF(KA.GT.KMIN.OR.Z(I).LT.0.1)GOTO 90
                KMIN=KA
                IL=I
90      CONTINUE
        IF(IL.NE.0)GOTO 110
        WRITE(6,100)
100     FORMAT(/,10X,'NO L-PHASE SOLVENT FOUND. INPUT IL')
        READ(6,*)IL
110     RETURN
        END
```

```
      SUBROUTINE BINGAM(IA,IB,GA,GB)

c     BINGAM contains the binary form of the NRTL
c     equation. Subroutine Solvnt utilizes this to
c     find the pair of system solvents.
c     ****************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
c     ****************************************
      DATA R/1.987/,REC/0.020408163/
      IF(IA.NE.IB.AND.IA.GT.0.AND.IB.GT.0.AND
     &  .IA.LE.N.AND.IB.LE.N)
     &  GOTO10
      WRITE(6,5)IA,IB
5     FORMAT(/,10X,'ILLEGAL INPUT TO SUBR.
     &  BINGAM-IA=',I6,'IB=',I6,/)
      STOP
10    TAB=C(IA,IB)/R/T
      TBA=C(IB,IA)/R/T
      GAB=DEXP(-ALPHA(IA,IB)*TAB)
      GBA=DEXP(-ALPHA(IB,IA)*TBA)
      GA=DEXP((TBA*(GBA/(49.+GBA))**2)
     & +((TAB*GAB)/(1+49.*GAB)**2))
      GB=DEXP((TAB*(GAB/(REC+GAB))**2)
     & +((TBA*GBA)/(1+REC*GBA)**2))
      RETURN
      END
```

```
                SUBROUTINE KEY

c       The heavy key component (smallest p*)
c       and the light key component (largest p*)
c       are found for VLE systems.
c       ****************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/INIT/XLIM(10,10),IV,IL
        COMMON/PVAP/PSTAR(20)
        REAL*8 MAX,MIN
c       ****************************************
        IV=0
        IL=0
        MAX=1.
        DO 10 I=1,N
                IF(PSTAR(I).LT.MAX.OR.Z(I).LT.0.1)GOTO 10
                MAX=PSTAR(I)
                IV=I
10      CONTINUE
        MIN=10000.
        DO 20 I=1,N
                IF(PSTAR(I).GT.MIN.OR.Z(I).LT.0.1)GOTO 20
                MIN=PSTAR(I)
                IL=I
20      CONTINUE
        WRITE(6,*)'IV,IL',IV,IL
        RETURN
        END
```

```
                    SUBROUTINE INITIL

c       This subroutine initializes x(i),y(i),and
c       the vapor phase fraction, A for LLE
c       calculations.
c       ****************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     &  DGAMX(20,20),DGAMY(20,20)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/INIT/XLIM(10,10),IV,IL
        SX=0.
        SY=0.
        IFDER=0
        DO 10 I=1,N
           X(I)=0.
           Y(I)=0.
           IF  (I.EQ.IL.OR.I.EQ.IV) GOTO 5
c       ****************************************
c       Solute species are set equal to the feed
c       fractions.

           X(I)=Z(I)
           Y(I)=Z(I)
c       ****************************************
c       XLIM is the estimated solubility.

           IF (XLIM(I,IL).LT.Z(I)) X(I)=0.75*XLIM(I,IL)
           IF (XLIM(I,IV).LT.Z(I)) Y(I)=0.75*XLIM(I,IV)

5          SX=SX+X(I)
           SY=SY+Y(I)
10      CONTINUE
c       ****************************************
        X(IV)=Z(IV)
        Y(IL)=Z(IL)
        IF(XLIM(IV,IL).LT.Z(IV)) X(IV)=0.75*XLIM(IV,IL)
        IF(XLIM(IL,IV).LT.Z(IL)) Y(IL)=0.75*XLIM(IL,IV)
        X(IL) = 1-(SX+X(IV))
        Y(IV) = 1-(SY+Y(IL))
        CALL NRTL(X,GAMX,DGAMX)
        CALL NRTL(Y,GAMY,DGAMY)

        DO 15 I=1,N
           RK(I)=GAMX(I)/GAMY(I)
15      CONTINUE
c       ****************************************
        A=(1.-Z(IL)+Z(IV))/2.
```

```
      DO 25 J=1,50
         F=0.
         DF=0.
         DO 20 I=1,N
            F=F+(Z(I)*(RK(I)-1.))/(A*(RK(I)-1.)+1.)
            DF=DF-Z(I)*((RK(I)-1.)/(A*(RK(I)-1.)+1.))**2
20       CONTINUE
         IF (DABS(F).LT.1.E-05) GO TO 35
         AN=A-F/DF
         IF (AN.GT.-1./(RK(IL)-1.))
     &   AN=A+((1./RK(IL)-1.)+A)/2.
         IF (AN.LT.-1./(RK(IV)-1.))
     &   AN=A+((1./RK(IV)-1.)+A)/2.
         A=AN
25    CONTINUE

      WRITE (6,30)
30    FORMAT(/,10X,'N-R ITERATION FOR
     & V/F BLEW IN INITIALIZATION')
      STOP

35    DO 40 I =1,N
         X(I)=Z(I)/(A*(RK(I)-1.)+1.)
         Y(I)=RK(I)*X(I)
40    CONTINUE
      RETURN
      END
```

```
                SUBROUTINE INITIV

c       Subroutine INITIV initializes x(i),y(i),
c       and the vapor phase fraction,A for VLE
c       calculations.
c       ****************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     &  DGAMX(20,20),DGAMY(20,20)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/INIT/XLIM(10,10),IV,IL
c       ****************************************
        DATA F1,F11,F2,F22,F3,F33,F4,F44,A
     &      /0.,0.,0.,0.,0.,0.,0.,0.,0./
        SX=0.
        SY=0.
        DO 5 I=1,N
           X(I)=0.
           Y(I)=0.
           X(I)=Z(I)
           Y(I)=Z(I)
           SX=SX+X(I)
           SY=SY+Y(I)
5       CONTINUE
        DO 10 I=1,N
                X(I)=X(I)/SX
                Y(I)=Y(I)/SY
10      CONTINUE

c       ****************************************
c       Activity coefficients are obtained.

        CALL NRTL (X,GAMX,DGAMX)
        CALL GAMVAP(Y,GAMY,DGAMY)
        DO 15 I=1,N
           RK(I) = GAMX(I)/GAMY(I)
15      CONTINUE
c       ****************************************
        DO 20 I=1,N
        F11=F11+RK(I)*Z(I)
        F22=F22+(RK(I)-1.)*Z(I)/(1.+RK(I))
        F33=F33+Z(I)/(1.+RK(I))
        F44=F44+Z(I)/RK(I)
20      CONTINUE
        F1=F11-1.
        F2=F22*2.
        F3=2.*F33-1.
        F4=F44-1.
```

```
C       ****************************************
        F=F2/(F2-F1)
        IF(F.GT.0.)A=.5*(1.-F)
        IF(F.LE.0.)A=(F3-.5*F4)/(F3-F4)
        IF(A.LT.0.05.OR.A.GT.0.95)A=.5
        RETURN
        END
```

```
      SUBROUTINE SSUBST(NSS,EPS,KIND,NLA2)
c       Phase equilibrium calculations are performed
c       using the successive substitution (previous
c       composition) algorithm.
c       ***********************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     &  GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     &  YPTS(50,10),DPTS(50,3)
        COMMON/COST/KOUNT
        COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     &  DGAMX(20,20),DGAMY(20,20)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/INIT/XLIM(10,10),IV,IL
        COMMON/POINTS/NPNTS
        COMMON/UPDATE/INXTX,INXTY
c       ***************************************
c       If the linearized real Taylor series
c       activity coefficient correlation (LTS) is
c       to be used, then a counter is initialized.

        IF(NLA2.EQ.0)GOTO 2
        INXTX=1
        INXTY=1
        NPNTS=0
c       ***************************************
2       EPS=DABS(EPS)
        IFDER=0
        ITER=0
        SX=0.
        SY=0.
c       ***************************************
c       Normalize phase compositions.

        DO 5 I=1,N
                IF(X(I).LT.0.)X(I)=0.
                IF(Y(I).LT.0.)Y(I)=0.
                SX=SX+X(I)
                SY=SY+Y(I)
5       CONTINUE
        DO 10 I=1,N
                X(I)=X(I)/SX
                Y(I)=Y(I)/SY
10      CONTINUE
c       ***************************************
c       Obtain activity coefficients from the
c       NRTL correlation or from the Antoine
```

```
c       equation.
                    CALL NRTL (X,GAMX,DGAMX)
        IF(KIND.EQ.1)CALL NRTL (Y,GAMY,DGAMY)
        IF(KIND.EQ.2)CALL GAMVAP(Y,GAMY,DGAMY)
c       ****************************************
c       Store data for later use in
c       creating a local model. These will
c       be accessed when the real linearized
c       Taylor series of the two-suffix Margules
c       equation (LTS) is created.

        IF(NLA2.EQ.0)GOTO 14
        CALL REGPNTS(X,GAMX,1)
        CALL REGPNTS(Y,GAMY,2)
c       ****************************************
14      DO 15 I=1,N
                RK(I)=GAMX(I)/GAMY(I)
15      CONTINUE
c       ****************************************
c       Start outer iterative loop.
c
        LIM=NSS
        IF(NSS.EQ.99)LIM=100
20      ITER=ITER+1
        KOUNT=KOUNT+2*N
        AOLD=A
        IF(ITER.GT.LIM)GO TO 100
c       ****************************************
c       Solve the Rachford Rice criterion
c       holding K values constant by a
c       univariate Newton-Raphson iteration.

        DO 30 M=1,50
                F=0
                DF=0
        DO 25 I=1,N
                RK1=RK(I)-1.
                RK2=A*RK1+1.
                F=F+Z(I)*RK1/RK2
                DF=DF-Z(I)*(RK1/RK2)**2
25              CONTINUE
        IF(DABS(F).LT.1.E-05)GO TO 40
        AN=A-F/DF
c       ****************************************
c       Limit A(V/F) to the poles of the Rachford-
c       Rice function.

        IF(KIND.EQ.2)GOTO 29
        IF(AN.GE.-1./(RK(IL)-1.))
```

```
     &  AN=A+(1./(RK(IL)-1.)+A)/2.
        IF(AN.LE.-1./(RK(IV)-1.))
     &  AN=A+(1./(RK(IV)-1.)+A)/2.
29      A=AN
30      CONTINUE
c       ****************************************
        WRITE(6,35)
        WRITE(66,35)
35      FORMAT(//,10X,'INNER LOOP BLEW IN SSUBST')
        STOP
c       ****************************************
c       Upon successful convergence of inner loop,
c       update iteration variables

40      A=AN
        SX=0.
        SY=0.
        DO 45 I=1,N
                X(I)=Z(I)/(A*(RK(I)-1.)+1.)
                Y(I)=RK(I)*X(I)
                IF (X(I).LT.0.)X(I)=0.
                IF (Y(I).LT.0.)Y(I)=0.
                SX=SX+X(I)
                SY=SY+Y(I)
45      CONTINUE
        DO 50 I=1,N
                X(I)=X(I)/SX
                Y(I)=Y(I)/SY
50      CONTINUE

                    CALL NRTL (X,GAMX,DGAMX)
        IF(KIND.EQ.1)CALL NRTL (Y,GAMY,DGAMY)
        IF(KIND.EQ.2)CALL GAMVAP(Y,GAMY,DGAMY)
c       ****************************************
        IF(NLA2.EQ.0)GOTO 54
        CALL REGPNTS(X,GAMX,1)
        CALL REGPNTS(Y,GAMY,2)
c       ****************************************
54      DO 55 I=1,N
                RK(I)=GAMX(I)/GAMY(I)
55      CONTINUE
c       ****************************************
c       Test for convergence,and, except in early
c       iterations, closeness to plait point and
c       divergence.
c
        IF(KIND.EQ.2)GOTO 60
        IF(ITER.LE.5)GOTO 60
        PPI=RK(IL)/RK(IV) + RK(IV)/RK(IL)
```

```
      IF (PPI.GT.10.OR.SS.GT.0.05)GOTO 60
      IF(ITER.GT.20.AND.PPI.LT.7.)GOTO 75
60    SS=0.
      DO 65 I=1,N
            SS=SS+DABS(GAMX(I)*X(I)-GAMY(I)*Y(I))
65    CONTINUE
C     ****************************************
      IF (SS.LT.EPS)GOTO 95
      IF (A.GE.0.AND.A.LE.1.)GOTO 20
      IF(ITER.LE.5.OR.SS.GE.0.05)GOTO 85
      GOTO 20
75    WRITE(6,80)
80    FORMAT(//,10X,'TOO CLOSE TO PLAIT POINT')
      STOP
85    WRITE(6,90)
90    FORMAT(//,10X,'A IS OUTSIDE
C     RANGE 0-1 AND GETTING WORSE')
C     STOP
      GOTO 96
95    CALL PRVALS(1,ITER,SS)
96    RETURN
100   IF(ITER.GT.50)GOTO 110
      RETURN
110   WRITE(6,120)
120   FORMAT(//,10X,'SSUBST FAILED
     & TO CONVERGE IN 50 ITERATIONS')
      STOP
      END
```

```
      SUBROUTINE REGPNTS(FR,AC,PH)

c     Phase fractions and activity coefficients
c     are stored in xpts,ypts and gxpts for
c     later use in the creation of the
c     real linearized Taylor series of the
c     two-suufix Margules (LTS model).
c     ****************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     & GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     & YPTS(50,10),DPTS(50,3)
      COMMON/DLHS/DGXPTS(50,10),DGYPTS(50,10)
      COMMON/UPDATE/INXTX,INXTY
      COMMON/POINTS/NPNTS
      REAL*8 FR(20),AC(20)
      INTEGER PH
c     ****************************************
      DATA R/1.987/
      DO 5 I=1,N
            IF(PH.EQ.1)XPTS(INXTX,I)=FR(I)
            IF(PH.EQ.2)YPTS(INXTY,I)=FR(I)
            IF(PH.EQ.1)DGXPTS(INXTX,I)=R*T*DLOG(AC(I))
            IF(PH.EQ.2)DGYPTS(INXTY,I)=R*T*DLOG(AC(I))
5     CONTINUE
      IF(PH.EQ.1)NPNTS=INXTX
      IF(PH.EQ.2)NPNTS=INXTY
      IF(PH.EQ.1)INXTX=INXTX+1
      IF(PH.EQ.2)INXTY=INXTY+1
      RETURN
      END
```

```
          SUBROUTINE NRTL(X,GAM,DGAM)

c       The Non-random Two-liquid activity
c       coefficient correlation represents
c       the complex molecular thermodynamic
c       (MT) model. When IFDER is 1,
c       activity coefficients and their
c       composition derivatives are provided.
c       ****************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        REAL*8 TAU(20,20),G(20,20),X(20),GAM(20)
     &  ,A(20),B(20),DB(20,20),DGAM(20,20)
c       ****************************************

        DATA R/1.987/
        DO 10 J=1,N
                DO 5 I=1,N
                        TAU(J,I)=C(J,I)/R/T
                        G(J,I)=DEXP(-ALPHA(J,I)*TAU(J,I))
5               CONTINUE
10      CONTINUE
        DO 20 J=1,N
                A(J)=0.
                B(J)=0.
                        DO 15 K=1,N
                                AK=G(K,J)*X(K)
                                A(J)=A(J)+AK
                                B(J)=B(J)+AK*TAU(K,J)
15                      CONTINUE
                B(J)=B(J)/A(J)
20      CONTINUE
        DO 30 J=1,N
               DO 25 K=1,N
                 DB(J,K)=G(K,J)*(TAU(K,J)-B(J))/A(J)
25             CONTINUE
30      CONTINUE
        DO 50 I=1,N
                GAM(I)=B(I)
                DO 40 J=1,N
                GAM(I)=GAM(I)+X(J)*DB(J,I)
                IF (IFDER.EQ.0)GOTO 40
                DGAM(I,J)=DB(I,J)+DB(J,I)
                   DO 35 K=1,N
                    DGAM(I,J)=DGAM(I,J)-X(K)*(G(J,K)*DB(K,I)
     &              +G(I,K)*DB(K,J))/A(K)
35                 CONTINUE
40      CONTINUE
```

```
      GAM(I)=DEXP(GAM(I))
      IF(IFDER.EQ.0)GOTO 50
      DO 45 J=1,N
            DGAM(I,J)=GAM(I)*DGAM(I,J)
45    CONTINUE
50    CONTINUE
      RETURN
      END
```

```
                SUBROUTINE GAMVAP (X,GAM,DGAM)

c       For VLE calculations, the activity coefficient
c       of each component in the vapor phase is
c       calculated as the ratio of the total pressure
c       to the vapor pressure.
c       ****************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 TAU(20,20),G(20,20),X(20),GAM(20),A(20)
     &  ,B(20),DB(20,20),DGAM(20,20)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/PVAP/PSTAR(20)
c       ****************************************

        DO 10 I=1,N
                GAM(I)=P/PSTAR(I)
10      CONTINUE

        DO 30 I=1,N
        DO 20 J=1,N
                DGAM(I,J)=0.
20      CONTINUE
30      CONTINUE
        RETURN
        END
```

```
                SUBROUTINE PRVALS(IM,ITER,OB)

c       The results of the iteration calculations
c       are printed to the screen or a data file.
c       ****************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     &  DGAMX(20,20),DGAMY(20,20)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/INIT/XLIM(10,10),IV,IL
        COMMON/SHOT/IFILE
        COMMON/CHARACTER/NAME(20)
c       OPEN(FILE='DUMP',UNIT=75,STATUS='UNKNOWN')
c       ****************************************
        IF(IM.EQ.1) WRITE(6,5)ITER
5       FORMAT(////,10X,'AFTER  ',I2,'
     &  SS ITERATIONS___',//)
        IF(IM.EQ.2) WRITE(6,10)ITER
10      FORMAT(////,10X,'AFTER  ',I2,'
     &  MV ITERATIONS___',//)
        IF(IM.EQ.3) WRITE(6,15)ITER
15      FORMAT(////,10X,'AFTER  ',I2,'
     &  LA ITERATIONS___',//)
        WRITE(6,20)
20      FORMAT(6X,'I',4X,'NAME',8X,'Z(I)'
     & ,5X,'Y(I)',5X,'X(I)',7X,'K(I)',
     &  5X,'GAMX(I)',3X,'GAMY(I)')
        DO 30 I=1,N
        II=2*I
        III=II-1
        WRITE(6,25)I,NAME(III),NAME(II),Z(I)
     & ,Y(I),X(I),RK(I),
     &           GAMX(I),GAMY(I)
25      FORMAT(4X,I3,2X,2A4,1X,3(2X,F7.3),3(2X,1PE9.2))
30      CONTINUE
        WRITE(6,35)T,A,OB
35      FORMAT(//,10X,'T=  ',F8.2,'K',10X,
     &  'A=  ',F8.4,' ERR= ',F15.7)
        WRITE(6,40)
40      FORMAT(//,10X,'ACTIVITY COEFFICIENT
     &  COMPOSITION DERIVATIVES',/)
        WRITE(6,45)
45      FORMAT(/,10X,'LIQUID PHASE',//)
        WRITE(6,50)(I,I=1,6)
50      FORMAT(7X,6(5X,I1,5X),/)
        DO 60 IROW=1,6
             WRITE(6,55)IROW,(DGAMX(IROW,ICOL),ICOL=1,6)
55           FORMAT(2X,I1,2X,6(F10.2,1X))
```

```
60       CONTINUE
         WRITE(6,65)
65       FORMAT(/,10X,'VAPOR PHASE',//)
         WRITE(6,70)(I,I=1,6)
70       FORMAT(7X,6(5X,I1,5X),/)
         DO 80 IROW=1,6
              WRITE(6,75)IROW,(DGAMY(IROW,ICOL),ICOL=1,6)
75            FORMAT(2X,I1,2X,6(F10.2,1X))
80       CONTINUE
         WRITE(6,*)
         RETURN
         END
```

```
      SUBROUTINE NRMT(NMT,ISIZE,DUM,DUMINV,KIND)

c     The multivariable Newton-Raphson approach
c     is implemented. Activity coefficients and
c     their composition derivatives are provided
c     by the NRTL equation.
c     ****************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     & DGAMX(20,20),DGAMY(20,20)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/INIT/XLIM(10,10),IV,IL
      COMMON/NR/DF(21,21),F(21),FN(21),XX(21)
      REAL*8 DUM(ISIZE,ISIZE),DUMINV(ISIZE,ISIZE),WKAREA(600)
      REAL*8 XN(20),YN(20)
c     ****************************************
      IFDER=1

c     ****************************************
c     The vector of independent variables is
c     set up.

      XX(ISIZE)=A
      DO 5 I=1,N
            II=I+N
            XX(I)=X(I)
            XX(II)=Y(I)
5     CONTINUE
      ITER=0
10    ITER=ITER+1
      SX=0.
      SY=0.
      DO 15 I=1,N
            XN(I)=X(I)
            YN(I)=Y(I)
            IF(XN(I).LT.0.)XN(I)=0.
            IF(YN(I).LT.0.)YN(I)=0.
            SX=SX+XN(I)
            SY=SY+YN(I)
15    CONTINUE
      DO 20 I=1,N
            XN(I)=XN(I)/SX
            YN(I)=YN(I)/SY
20    CONTINUE
                  CALL NRTL(XN,GAMX,DGAMX)
      IF(KIND.EQ.1)CALL NRTL(YN,GAMY,DGAMY)
      IF(KIND.EQ.2)CALL GAMVAP(YN,GAMY,DGAMY)
      DO 25 I=1,N
```

```
                RK(I)=GAMX(I)/GAMY(I)
25          CONTINUE
c           ****************************************
c           The Jacobian and the vector of defining
c           equations are set up.

            DO 35 I=1,ISIZE
                    DO 30 J=1,ISIZE
                       DF(I,J)=0.
                       IF(I.NE.ISIZE.OR.J.EQ.ISIZE)GOTO 30
                       DF(I,J)=-1.
                       IF(J.GT.N)DF(I,J)=1.
30                  CONTINUE
35          CONTINUE
            F(ISIZE)=0.
            DO 45 I=1,N
                    II=I+N
                    F(I)=Z(I)-A*XX(II)+(A-1.)*XX(I)
                    F(II)=GAMX(I)*XX(I)-GAMY(I)*XX(II)
                    F(ISIZE)=F(ISIZE)+XX(II)-XX(I)
                    DF(I,I)=A-1.
                    DF(I,II)=-A
                    DF(I,ISIZE)=XX(I)-XX(II)
                    DO 40 J=1,N
                            JJ=J+N
                            DF(II,J)=XX(I)*DGAMX(I,J)
                            DF(II,JJ)=-XX(II)*DGAMY(I,J)
                            IF(I.NE.J)GOTO 40
                            DF(II,J)=DF(II,J)+GAMX(I)
                            DF(II,JJ)=DF(II,JJ)-GAMY(I)
40                  CONTINUE
45          CONTINUE
            SF=0.
            DO 50 I=1,ISIZE
                    SF=SF+DABS(F(I))
50          CONTINUE
            DO 55 I=1,N
                    II=I+N
                    X(I)=XX(I)
                    Y(I)=XX(II)
55          CONTINUE
            IT=ITER-1
c           ****************************************
c           If the vector of defining equations is
c           less than some convergence criterion,
c           exit the subroutine.

            IF(SF.LT.5.E-05)GOTO 110
c           ****************************************
```

```
c       Invert the Jacobian using IMSL routine
c       LINV2F or subroutine MATINV and make the
c       iterative step.

        DO 70 I=1,ISIZE
                DO 65 J=1,ISIZE
                        DUM(I,J)=DF(I,J)
65              CONTINUE
70      CONTINUE
c       CALL MATINV(DUM,ISIZE,ISIZE)
        CALL LINV2F(DUM,ISIZE,ISIZE,DUMINV,-1,WKAREA,IER)
        DO 80 I=1,ISIZE
                DO 75 J=1,ISIZE
                        DF(I,J)=DUMINV(I,J)
75              CONTINUE
80      CONTINUE
        DO 90 I=1,ISIZE
                FN(I)=0.
                DO 85 J=1,ISIZE
                        FN(I)=FN(I)+F(J)*DF(I,J)
85              CONTINUE
90      CONTINUE
        DO 95 I=1,ISIZE
                XX(I)=XX(I)-FN(I)
95      CONTINUE
        A=XX(ISIZE)
        DO 100 I=1,N
                II=I+N
                X(I)=XX(I)
                Y(I)=XX(II)
100     CONTINUE
        IF(ITER.EQ.NMT)RETURN
        IF(ITER.LE.30)GOTO 10
c       ****************************************
        WRITE(6,105)
105     FORMAT(/,10X,'NR-II FAILED TO CONVERGE')
        STOP
110     CALL PRVALS(2,IT,SF)
        RETURN
        END
```

```
                SUBROUTINE MATINV(A,N,ND)

c       Subroutine to invert a general matrix.
c       ****************************************
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 A(ND,ND),AINV(21,21),WKAREA1(30),WKAREA2(30)
        REAL*8 IDENT,MULT
        INTEGER ROW,PIVROW,ORDER
c       ****************************************
        DO 10 I = 1,N
           DO 5 J = 1,N
              AINV(I,J) = 0.0
              AINV(I,I) = 1.0
5          CONTINUE
10      CONTINUE
        ROW = 1
30      CALL FINDPV(A,N,ROW,PIVOT,PIVROW)
        IF (PIVOT.LE.1.0D-15) THEN
           WRITE (6,*) 'ZERO PIVOT FOUND--YOUR MATRIX IS SINGULAR.'
           RETURN
        ENDIF
        DO 35 I = 1,N
           WKAREA1(I) = A(ROW,I)
           WKAREA2(I) = AINV(ROW,I)
           A(ROW,I) = A(PIVROW,I)
           AINV(ROW,I) = AINV(PIVROW,I)
           A(PIVROW,I) = WKAREA1(I)
           AINV(PIVROW,I) = WKAREA2(I)
35      CONTINUE
        DIVIDE = A(ROW,ROW)
        DO 40 I = 1,N
           A(ROW,I) = A(ROW,I)/DIVIDE
           AINV(ROW,I) = AINV(ROW,I)/DIVIDE
40      CONTINUE
        DO 50 I = 1,N
           MULT = A(I,ROW)
           DO 45 K = 1,N
              IF (I .NE. ROW) THEN
                 A(I,K) = A(I,K) - MULT*A(ROW,K)
                 AINV(I,K) = AINV(I,K) - MULT*AINV(ROW,K)
              ENDIF
45         CONTINUE
50      CONTINUE
        ROW = ROW + 1
        IF (ROW .LE. N) GO TO 30
        DO 60 I = 1,N
           DO 55 J = 1,N
              A(I,J) = AINV(I,J)
55            CONTINUE
```

```
60      CONTINUE
        RETURN
        END

        SUBROUTINE FINDPV(A,M,LEVEL,PIVOT,PIVROW)

c       Subroutine to find the maximum pivot element.
c       This is used in conjunction with MATINV.
c       ****************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        DIMENSION A(M,M)
        INTEGER PIVROW
c       ****************************************
        PIVOT =DABS(A(LEVEL,LEVEL))
        PIVROW = LEVEL
        IF (LEVEL.EQ.M) RETURN
        DO 5 I = LEVEL,M
           IF (DABS(A(I,LEVEL)) .GT. PIVOT) THEN
              PIVOT = DABS(A(I,LEVEL))
              PIVROW = I
           ENDIF
5       ENDDO
        RETURN
        END
```

```
      SUBROUTINE NRLA1
     &  (NLA,MODEL,ISIZE,DUM,DUMINV,KIND,NUM)

c     Phase equilibrium calculations are performed by
c     the multivariable Newton-Raphson technique.
c     A local approximation will replace the NRTL
c     equation for most of the calculations.
c     ****************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BASEACT/XBGAM(20,20),YBGAM(20,20)
      COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     &  DGAMX(20,20),DGAMY(20,20)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/INIT/XLIM(10,10),IV,IL
      COMMON/NR/DF(21,21),F(21),FN(21),XX(21)
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     &  GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     &  YPTS(50,10),DPTS(50,3)
      REAL*8 DUM(ISIZE,ISIZE),DUMINV(ISIZE,ISIZE)
     &  ,WKAREA(600)
      REAL*8 XN(20),YN(20),LIMIT
      REAL*8 DERGAMX(20,20),DERGAMY(20,20)
c     ****************************************
c     Pseudo-composition definitions for the LA
c     models must be devised

      IF(MODEL.EQ.1.OR.MODEL.EQ.3)CALL PSEUDOTERN
      IF(MODEL.EQ.2.OR.MODEL.EQ.4)CALL PSEUDOQUAT
c     ****************************************
c     The moving base point strategy will be used with
c     either of the expansion models if IMBP=1

      IMBP=0
      MP=0
      IF(MODEL.EQ.3.OR.MODEL.EQ.4)IMBP=1
c     ****************************************
c     Create the local approximation model.

      CALL INITLA1(NUM,MODEL,KIND,0)
c     ****************************************
      DO 2 I=1,ISIZE
            XX(I)=0.
2     CONTINUE
      XX(ISIZE)=A
      DO 5 I=1,N
            II=I+N
            XX(I)=X(I)
            XX(II)=Y(I)
```

```
                XN(I)=0.
                YN(I)=0.
5       CONTINUE

        LCNVD=0
        ITER=0
10      ITER=ITER+1
c       ****************************************
c       Normalize the phase compositions.
        SX=0.
        SY=0.
        DO 15 I=1,N
                XN(I)=X(I)
                YN(I)=Y(I)
                IF(XN(I).LT.0.)XN(I)=0.
                IF(YN(I).LT.0.)YN(I)=0.
                SX=SX+XN(I)
                SY=SY+YN(I)
15      CONTINUE
        DO 21 I=1,N
                XN(I)=XN(I)/SX
                YN(I)=YN(I)/SY
21      CONTINUE

c       ****************************************
c       Activity coefficients are now obtained
c       from the appropriate local approximation.
c       Get heavy phase activity coefficients
c       and their composition derivatives.

        IF(MODEL.EQ.1)CALL LATERN(XN,GAMX,DGAMX
     &  ,IPSX,LCNVD)
        IF(MODEL.EQ.2)CALL LAQUAT(XN,GAMX,DGAMX
     &  ,IPSX,LCNVD)
        IF(MODEL.EQ.3)CALL LAEXPTER(XN,GAMX,DGAMX,IPSX
     &  ,LCNVD,XBGAM)
        IF(MODEL.EQ.4)CALL LAEXPQUAT(XN,GAMX,DGAMX,IPSX
     &  ,LCNVD,XBGAM)

c       Get light phase activity coefficients.If
c       the light phase is a vapor phase, assume
c       ideality and obtain an expression for
c       the activity coefficient from vapor pressure
c       information.

        IF(KIND.EQ.1.AND.MODEL.EQ.1)
     &          CALL LATERN(YN,GAMY,DGAMY,IPSY,LCNVD)
        IF(KIND.EQ.1.AND.MODEL.EQ.2)
     &          CALL LAQUAT(YN,GAMY,DGAMY,IPSY,LCNVD)
```

```
      IF(KIND.EQ.1.AND.MODEL.EQ.3)
     &       CALL LAEXPTER(YN,GAMY,DGAMY,IPSY,LCNVD,YBGAM)

      IF(KIND.EQ.1.AND.MODEL.EQ.4)
     &       CALL LAEXPQUAT(YN,GAMY,DGAMY,IPSY,LCNVD,YBGAM)

      IF(KIND.EQ.2)CALL GAMVAP(YN,GAMY,DGAMY)
C     ****************************************
C     Compute equilibrium ratios and set up the defining
C     equations  vector F and the Jacobian DF

22    DO 25 I=1,N
            RK(I)=GAMX(I)/GAMY(I)
25    CONTINUE

      DO 35 I=1,ISIZE
            DO 30 J=1,ISIZE
             DF(I,J)=0.
             IF(I.NE.ISIZE.OR.J.EQ.ISIZE)GO TO 30
             DF(I,J)=-1.
             IF(J.GT.N)DF(I,J)=1.
30          CONTINUE
35    CONTINUE
      F(ISIZE)=0.

      DO 45 I=1,N
            II=I+N
            F(I)=Z(I)-A*XX(II)+(A-1.)*XX(I)
            F(II)=GAMX(I)*XX(I)-GAMY(I)*XX(II)
            F(ISIZE)=F(ISIZE)+XX(II)-XX(I)
            DF(I,I)=A-1.
            DF(I,II)=-A
            DF(I,ISIZE)=XX(I)-XX(II)
                  DO 40 J=1,N
                     JJ=J+N
                     DF(II,J)=XX(I)*DGAMX(I,J)
                     DF(II,JJ)=-XX(II)*DGAMY(I,J)
                     IF(I.NE.J)GO TO 40
                     DF(II,J)=DF(II,J)+GAMX(I)
                     DF(II,JJ)=DF(II,JJ)-GAMY(I)
40                CONTINUE
45    CONTINUE
      SF=0.
      DO 50 I=1,ISIZE
            SF=SF+DABS(F(I))
50    CONTINUE
      DO 55 I=1,N
            II=I+N
```

```
                X(I)=XX(I)
                Y(I)=XX(II)
55          CONTINUE

            IT=ITER-1

c           ****************************************
            WRITE(6,60)IT,SF
60          FORMAT(/,10X,'AFTER ',I2,'
     &      LA ITERATIONS, ERR= ',F15.8)
            LIMIT=5.E-05
c           ****************************************
c           If MODEL = 1 or MODEL = 2 : let the
c           LA model converge to its own solution.
c           LCNVD is set to 1 and activity
c           coefficients are then obtained from the
c           NRTL equation. After convergence, exit
c           the subroutine.

            IF(SF.LT.LIMIT.AND.LCNVD.EQ.1)GOTO110
            IF(SF.LT.LIMIT.AND.IMBP.EQ.0)LCNVD = 1
c           ****************************************
c           If MODEL = 2 or MODEL = 3 : When
c           the LA model has converged to its
c           solution , update the base point
c           (around which the model has been
c           expanded) with activity coefficients
c           provided by the NRTL model.

            IF(SF.LT.LIMIT.AND.MP.EQ.20)GOTO110
            IF(SF.LT.LIMIT.AND.IMBP.EQ.1)THEN
                    CALL INITLA1(NUM,MODEL,KIND,1)
                    MP=MP+1
            END IF

c           ****************************************
c           Make the iterative step X=X-(DF^-1)(F)

            DO 70 I=1,ISIZE
                    DO 65 J=1,ISIZE
                            DUM(I,J)=DF(I,J)
65                  CONTINUE
70          CONTINUE
c           CALL MATINV(DUM,ISIZE,ISIZE)
            CALL LINV2F(DUM,ISIZE,ISIZE,DUMINV,-1,WKAREA,IER)
            DO 80 I=1,ISIZE
                    DO 75 J=1,ISIZE
                            DF(I,J)=DUMINV(I,J)
75                  CONTINUE
```

```
80      CONTINUE
        DO 90 I=1,ISIZE
                FN(I)=0.
                DO 85 J=1,ISIZE
                        FN(I)=FN(I)+F(J)*DF(I,J)
85              CONTINUE
90      CONTINUE
        DO 95 I=1,ISIZE
                XX(I)=XX(I)-FN(I)
95      CONTINUE
        A=XX(ISIZE)
        DO 100 I=1,N
                II=I+N
                X(I)=XX(I)
                Y(I)=XX(II)
100     CONTINUE
        IF(ITER.GT.100)RETURN
        IF(ITER.LE.100)GOTO10

        WRITE(6,105)
105     FORMAT(/,10X,'NR-II(LA) FAILED TO CONVERGE')
        STOP
110     CALL PRVALS(3,IT,SF)
        RETURN
        END
```

```
      SUBROUTINE NRLA2(NLA,ISIZE,DUM,DUMINV,KIND)

c     Phase equilibrium calculations are performed by the
c     multivariable Newton-Raphson technique. A linearized
c     Taylor-series expansion (LTS) will replace the NRTL
c     equation for most of the calculations.
c     ****************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BASEACT/XBGAM(20,20),YBGAM(20,20)
      COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20)
     &GAMY(20),DGAMX(20,20),DGAMY(20,20)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/INIT/XLIM(10,10),IV,IL
      COMMON/NR/DF(21,21),F(21),FN(21),XX(21)
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     &GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     &YPTS(50,10),DPTS(50,3)
      COMMON/POINTS/NPNTS
      REAL*8 DUM(ISIZE,ISIZE)
     &,DUMINV(ISIZE,ISIZE),WKAREA(600)
      REAL*8 XN(20),YN(20),LIMIT
c     ****************************************
c     The number of parameters,(NUM), in the LA
c     model is equal to the number of components
c     in the mixture.

      IF(N.EQ.3)NUM=3
      IF(N.EQ.4)NUM=4
      IF(N.EQ.5)NUM=5
      IF(N.EQ.6)NUM=6

      MP=0
c     ****************************************
c     Create the local approximation model.

      CALL INITLA2(NUM,KIND,0)
c     ****************************************
      XX(ISIZE)=A
      DO 5 I=1,N
            II=I+N
            XX(I)=X(I)
            XX(II)=Y(I)
            XN(I)=0.
            YN(I)=0.
5     CONTINUE

      LCNVD=0
      ITER=0
```

```
10      ITER=ITER+1
        SX=0.
        SY=0.
        DO 15 I=1,N
                XN(I)=X(I)
                YN(I)=Y(I)
                IF(XN(I).LT.0.)XN(I)=0.
                IF(YN(I).LT.0.)YN(I)=0.
                SX=SX+XN(I)
                SY=SY+YN(I)
15      CONTINUE
        DO 21 I=1,N
                XN(I)=XN(I)/SX
                YN(I)=YN(I)/SY
21      CONTINUE
c       ****************************************
c       Activity coefficients are now obtained from the
c       appropriate local approximation

c       Get heavy phase activity coefficients
c       and their composition derivatives.

        CALL LAMODEL(XN,GAMX,DGAMX,IPSX,LCNVD,XBGAM)

c       Get light phase activity coefficients and
c       compositions.

        CALL LAMODEL(YN,GAMY,DGAMY,IPSY,LCNVD,YBGAM)
c       ****************************************
c       Compute equilibrium ratios and set up the
c       defining equations  vector F and the Jacobian
c       DF

22      DO 25 I=1,N
                RK(I)=GAMX(I)/GAMY(I)
25      CONTINUE

        DO 35 I=1,ISIZE
                DO 30 J=1,ISIZE
                 DF(I,J)=0.
                 IF(I.NE.ISIZE.OR.J.EQ.ISIZE)GO TO 30
                 DF(I,J)=-1.
                 IF(J.GT.N)DF(I,J)=1.
30              CONTINUE
35      CONTINUE
        F(ISIZE)=0.
        DO 45 I=1,N
                II=I+N
                F(I)=Z(I)-A*XX(II)+(A-1.)*XX(I)
```

```
                F(II)=GAMX(I)*XX(I)-GAMY(I)*XX(II)
                F(ISIZE)=F(ISIZE)+XX(II)-XX(I)
                DF(I,I)=A-1.
                DF(I,II)=-A
                DF(I,ISIZE)=XX(I)-XX(II)
                        DO 40 J=1,N
                         JJ=J+N
                         DF(II,J)=XX(I)*DGAMX(I,J)
                         DF(II,JJ)=-XX(II)*DGAMY(I,J)
                         IF(I.NE.J)GO TO 40
                         DF(II,J)=DF(II,J)+GAMX(I)
                         DF(II,JJ)=DF(II,JJ)-GAMY(I)
40                      CONTINUE
45      CONTINUE
        SF=0.
        DO 50 I=1,ISIZE
                SF=SF+DABS(F(I))
50      CONTINUE
        DO 55 I=1,N
                II=I+N
                X(I)=XX(I)
                Y(I)=XX(II)
55      CONTINUE

        IT=ITER-1
c       ****************************************
        WRITE(6,60)IT,SF
60      FORMAT(/,10X,'AFTER ',I2,' ITERATIONS, ERR= ',F15.8)
        WRITE(6,*)'A IS',A
c       ****************************************
c       After the initial model has converged,
c       update the base points with NRTL
c       activity coefficients.

        LIMIT=5.E-05
        IF(SF.LT.LIMIT.AND.MP.EQ.10)GOTO110
        IF(SF.LT.LIMIT)THEN
                CALL INITLA2(NUM,KIND,1)
                MP=MP+1
        END IF
c       ****************************************
c       Make the iterative step X=X-(DF^-1)(F)

63      DO 70 I=1,ISIZE
                DO 65 J=1,ISIZE
                        DUM(I,J)=DF(I,J)
65              CONTINUE
70      CONTINUE
```

```
        CALL LINV2F(DUM,ISIZE,ISIZE,DUMINV,-1,WKAREA,IER)
        DO 80 I=1,ISIZE
                DO 75 J=1,ISIZE
                        DF(I,J)=DUMINV(I,J)
75              CONTINUE
80      CONTINUE
        DO 90 I=1,ISIZE
                FN(I)=0.
                DO 85 J=1,ISIZE
                        FN(I)=FN(I)+F(J)*DF(I,J)
85              CONTINUE
90      CONTINUE
        DO 95 I=1,ISIZE
                XX(I)=XX(I)-FN(I)
95      CONTINUE
        A=XX(ISIZE)
        DO 100 I=1,N
                II=I+N
                X(I)=XX(I)
                Y(I)=XX(II)
100     CONTINUE
        IF(ITER.GT.100)RETURN
        IF(ITER.LE.100)GOTO10
        WRITE(6,105)
105     FORMAT(/,10X,'NR-II(LA) FAILED TO CONVERGE')
        STOP
110     CALL PRVALS(3,IT,SF)
        RETURN
        END
```

```
      SUBROUTINE DIFDER(XD,IPSD,PHASE)

c     Computes composition derivatives by
c     difference.
c     ****************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/CHARACTER/NAME(20)
      INTEGER PHASE,IPSD(10,10)
      REAL*8 XD(20),PGAM(20),PDGAM(20,20),SGAM(20)
      REAL*8 DERGAM(20,20),GAM(20)
      REAL*8 DGAM(20,20),NGAM(20),NDGAM(20,20)
      REAL*8 NGAMX(20),NGAMY(20),NDGAMX(20,20)
     &  ,NDGAMY(20,20)
      REAL*8 XDX(20),XDY(20),RK(20)
c     DX=.001
c     LCNVD=0
      IFDER=1
      CALL NRTL(XD,NGAM,NDGAM)
      DO 10 I=1,N
      IF(PHASE.EQ.1)NGAMX(I)=NGAM(I)
      IF(PHASE.EQ.2)NGAMY(I)=NGAM(I)
      IF(PHASE.EQ.1)XDX(I)=XD(I)
      IF(PHASE.EQ.2)XDY(I)=XD(I)
      RK(I)=0.
      DO 11 J=1,N
      IF(PHASE.EQ.1)NDGAMX(I,J)=NDGAM(I,J)
      IF(PHASE.EQ.2)NDGAMY(I,J)=NDGAM(I,J)
11    CONTINUE
10    CONTINUE
      IF(PHASE.EQ.1)GOTO 33
      WRITE(6,*)'NRTL DATA'
      WRITE(6,20)
20    FORMAT(6X,'I',4X,'NAME',8X,'Z(I)',5X,
     &  'Y(I)',5X,'X(I)',7X,'K(I)',
     &  5X,'GAMX(I)',3X,'GAMY(I)')
      DO 30 I=1,N
      II=2*I
      III=II-1
      WRITE(6,25)I,NAME(III),NAME(II),Z(I)
     &  ,XDY(I),XDX(I),RK(I),NGAMX(I),NGAMY(I)
25    FORMAT(4X,I3,2X,2A4,1X,3(2X,F7.3),3(2X,1PE9.2))
30    CONTINUE
      WRITE(6,37)
37    FORMAT(//,10X,'ACTIVITY COEFFICIENT
     &  COMPOSITION DERIVATIVES',/)
      WRITE(6,38)
38    FORMAT(/,10X,'LIQUID PHASE',//)
      WRITE(6,40)(I,I=1,6)
```

```
40      FORMAT(7X,6(5X,I1,5X),/)
        DO 45 IROW=1,6
              WRITE(6,50)IROW,(NDGAMX(IROW,ICOL),ICOL=1,6)
50            FORMAT(2X,I1,2X,6(F10.2,1X))
45      CONTINUE
        WRITE(6,46)
46      FORMAT(/,10X,'VAPOR PHASE',//)
        WRITE(6,55)(I,I=1,6)
55      FORMAT(7X,6(5X,I1,5X),/)
        DO 60 IROW=1,6
              WRITE(6,65)IROW,(NDGAMY(IROW,ICOL),ICOL=1,6)
65            FORMAT(2X,I1,2X,6(F10.2,1X))
60      CONTINUE
        WRITE(6,*)
C       ****************************************
C       DO 10 I=1,N
C             XD(I)=XD(I)+DX
C             CALL LATERN(XD,PGAM,PDGAM,IPSD,LCNVD)
C             SGAM(I)=0.
C             DO 20 J=1,N
C                   IF(I.EQ.J)GOTO 20
C                   DERGAM(J,I)=(PGAM(J)-GAM(J))/DX
C                   SGAM(I)=SGAM(I)-DERGAM(J,I)
c20           CONTINUE
C             DERGAM(I,I)=SGAM(I)
C             XD(I)=XD(I)-DX
c10     CONTINUE
C       DO 24 I=1,N
C       DO 34 J=1,N
C             DERGAM(I,J)=DERGAM(J,I)
c34     CONTINUE
c24     CONTINUE
C       DO 25 I=1,N
C             WRITE(6,*)
C             WRITE(6,*)'PHASE,COMP #',PHASE,I
C             WRITE(6,*)
C         WRITE(6,*)'ANALYTIC,NRTL A.COEFF',
C    &          I,GAM(I),NGAM(I)
C       DO 30 J=1,N
C         WRITE(6,*)'DIFFERENCE,ANALYTIC,NRTL DERIVATIVES',
C    &          I,J,DERGAM(I,J),DGAM(I,J),NDGAM(I,J)
c30     CONTINUE
c25     CONTINUE
33      RETURN
        END
```

SUBROUTINE PSEUDOQUAT

```
c       Pseudo composition definitions are set
c       for the pseudo-quartenary two-suffix
c       Margules equation.
c       ******************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     &  DGAMX(20,20),DGAMY(20,20)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/INIT/XLIM(10,10),IV,IL
        COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     &  GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     &  YPTS(50,10),DPTS(50,3)
        REAL*8 SOLX(20),SOLY(20)
        REAL*8 KMAX,KMIN
        INTEGER HIGH,LOW
c       ******************************************
        DO 10 I=1,N
        SOLX(I)=0.
        SOLY(I)=0.
10      CONTINUE

        DO 60 I=1,N
        DO 20 J=1,N
                IPSX(I,J)=0
                IF(I.EQ.J)GOTO 20
                IPSX(I,J)=3
                IF(J.EQ.IV.OR.RK(J).GT.RK(IV))IPSX(I,J)=1
                IF(J.EQ.IL.OR.RK(J).LT.RK(IL))IPSX(I,J)=2
                IF(J.NE.IV.AND.J.NE.IL)SOLX(J)=RK(J)
20      CONTINUE
        KMAX=0.
        KMIN=100.
        IF(I.NE.IV)GOTO 40
                DO 30 K=1,N
                IF(SOLX(K).LE.KMAX)GOTO 30
                KMAX=SOLX(K)
                HIGH=K
30              CONTINUE
                IPSX(I,HIGH)=1
40      IF(I.NE.IL) GOTO 60
                DO 50 K=1,N
                IF(SOLX(K).EQ.0..OR.SOLX(K).GT.KMIN)GOTO 50
                KMIN=SOLX(K)
                LOW=K
50              CONTINUE
                IPSX(I,LOW)=2
```

```
60      CONTINUE
        DO 120 I=1,N
70      DO 80 J=1,N
                IPSY(I,J)=0
                IF(I.EQ.J)GOTO 80
                IPSY(I,J)=3
                IF(J.EQ.IV.OR.RK(J).GT.RK(IV))IPSY(I,J)=1
                IF(J.EQ.IL.OR.RK(J).LT.RK(IL))IPSY(I,J)=2
                IF(J.NE.IV.AND.J.NE.IL)SOLY(J)=RK(J)
80      CONTINUE
        KMAX=0.
        KMIN=100.
        IF(I.NE.IV)GOTO 100
                DO 90 K=1,N
                IF(SOLY(K).LE.KMAX)GOTO 90
                KMAX=SOLY(K)
                HIGH=K
90              CONTINUE
                IPSY(IV,HIGH)=1
100     IF(I.NE.IL)GOTO 120
                DO 110 K=1,N
                IF(SOLY(K).EQ.0.OR.SOLY(K).GT.KMIN)GOTO 110
                KMIN=SOLY(K)
                LOW=K
110             CONTINUE
                IPSY(IL,LOW)=2
120     CONTINUE
        RETURN
        END
```

```
                SUBROUTINE PSEUDOTERN
c       Pseudo-components are defined for the
c       for the pseudo-ternary 2-suffix Margules.
c       ****************************************
        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     &  DGAMX(20,20),DGAMY(20,20)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/INIT/XLIM(10,10),IV,IL
        COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     &  GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     &  YPTS(50,10),DPTS(50,3)
c       ****************************************
        KG1=0
        DO 5 I=1,N
                IF(RK(I).GE.1.)KG1=KG1+1
5       CONTINUE
        KL1=N-KG1
        DO 20 I=1,N
                DO 15 J=1,N
                        IPSX(I,J)=0
                        IF(I.EQ.J)GOTO10
                        IPSX(I,J)=-1
                        IF(RK(J).GE.1.)IPSX(I,J)=1
        IF(I.EQ.IL.AND.KL1.LE.1.AND.J.EQ.IV)IPSX(I,J)=-1
                        IF(I.NE.IV)GOTO 10
                        IPSX(I,J)=-1
                        IF(J.EQ.IL)IPSX(I,J)=1
10                      IPSY(I,J)=0
                        IF(I.EQ.J)GOTO15
                        IPSY(I,J)=-1
                        IF(RK(J).GE.1.)IPSY(I,J)=1
        IF(I.EQ.IV.AND.KG1.LE.1.AND.J.EQ.IL)IPSY(I,J)=1
                        IF(I.NE.IL)GOTO15
                        IPSY(I,J)=-1
                        IF(J.EQ.IV)IPSY(I,J)=1
15              CONTINUE
20      CONTINUE
        IF(N.EQ.3)THEN
        IPSX(1,2)=-1
        IPSX(1,3)=1
        IPSX(2,1)=-1
        IPSX(2,3)=1
        IPSX(3,1)=-1
        IPSX(3,2)=1
        END IF
        RETURN
        END
        SUBROUTINE INITLA1(NUM,MODEL,KIND,UP)
```

```
c       INITLA1 provides the data points
c       that will be used in the LA
c       parameter regression. This will
c       be used in conjunction with NRLA1.
c       ****************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     &  DGAMX(20,20),DGAMY(20,20)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/INIT/XLIM(10,10),IV,IL
        COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     &  GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     &  YPTS(50,10),DPTS(50,3)
        COMMON/UPDATE/INXTX,INXTY
        COMMON/PERTA/PERT1,PERT2,PERT3,PERT4,PERT5
        INTEGER IPTS(50),UP,UPDA
        REAL*8 XTX(21,21),XTXI(21,21),LPTS(50,3)
        REAL*8 VPTS(50,3)
c       ****************************************
c       When one of the expansion models is
c       being used and needs to be updated,
c       new NRTL activity coefficients are
c       all that are needed.

        IF(UP.EQ.1)GOTO 40
c       ****************************************
c       The perturbation factors that will be
c       used in subroutine POINTS1 are set up

        DATA PERT1,PERT2,PERT3,PERT4,PERT5
     &       /.6,  .7,   .8,    .9,   .95/
c       ****************************************
c       The number of data sets that will be
c       used in the parameter regression is
c       set up.

        WRITE(6,5)
5       FORMAT(10X,'INPUT NUMBER OF REGRESSION
     &  DATA SETS')
        READ(6,*)NDPTS
c       ************************************************
c       Subroutine POINTS1 is now called to obtain
c       the NDPTS regression sets for each phase.

        CALL POINTS1(X,NDPTS,XPTS)
                DO 20 I=1,NDPTS
                        SX=0.0
```

```
                DO 10 J=1,N
                        IF(XPTS(I,J).LT.0.)XPTS(I,J)=.001
                        SX=SX+XPTS(I,J)
10              CONTINUE
                DO 15 J=1,N
                        XPTS(I,J)=XPTS(I,J)/SX
15              CONTINUE
20      CONTINUE

        IF(KIND.EQ.2)GOTO 40

        CALL POINTS1(Y,NDPTS,YPTS)
                DO 35 I=1,NDPTS
                        SY=0.
                DO 25 J=1,N
                        IF(YPTS(I,J).LT.0.)YPTS(I,J)=.001
                        SY=SY+YPTS(I,J)
25              CONTINUE
                DO 30 J=1,N
                YPTS(I,J)=YPTS(I,J)/SY
30              CONTINUE
35      CONTINUE
c       ****************************************
c       A least squares regression is now
c       performed to obtain the necessary
c       model parameters.

40      DO 45 I=1,NDPTS
                IPTS(I)=I
45      CONTINUE
50      UPDA=0
        IF(UP.EQ.1)UPDA=1
        IF(MODEL.EQ.1)CALL NEWMOD(IPTS,1,NDPTS,NUM,XTX,XTXI)
        IF(MODEL.EQ.2)
     &  CALL NEWMOD(IPTS,1,NDPTS,NUM,XTX,XTXI)
        IF(MODEL.EQ.3)
     &  CALL NMEXPTER(IPTS,1,NDPTS,NUM,XTX,XTXI,UPDA)
        IF(MODEL.EQ.4)
     &  CALL NMEXPQUAT(IPTS,1,NDPTS,NUM,XTX,XTXI,UPDA)

c       Get vapor phase activity coefficients

        IF(KIND.EQ.2)GOTO 55
        IF(MODEL.EQ.1)
     &          CALL NEWMOD(IPTS,2,NDPTS,NUM,XTX,XTXI)
        IF(MODEL.EQ.2)
     &       CALL NEWMOD(IPTS,2,NDPTS,NUM,XTX,XTXI)
        IF(MODEL.EQ.3)
     &       CALL NMEXPTER(IPTS,2,NDPTS,NUM,XTX,XTXI,UPDA)
```

```
      IF(MODEL.EQ.4)
     &      CALL NMEXPQUAT(IPTS,2,NDPTS,NUM,XTX,XTXI,UPDA)
55    RETURN
      END
```

```
      SUBROUTINE INITLA2(NUM,KIND,UP)

c     NMEXP is called to create or update
c     the parameters of the linearized Taylor
c     series form of the 2-suffix Margules (LTS).
c     ******************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/UPDATE/INXTX,INXTY
      COMMON/POINTS/NPNTS
      REAL*8 XTX(21,21),XTXI(21,21)
      INTEGER IPTS(50),UP

c     ******************************************
      DO 10 I=NPNTS
      IPTS(I)=I
10    CONTINUE

c     ******************************************
c     NMEXP will regress for the parameters
c     of a linearized Taylor series LA for
c     each component in each phase.

      CALL NMEXP(IPTS,1,NPNTS,NUM,XTX,XTXI,UP)
      CALL NMEXP(IPTS,2,NPNTS,NUM,XTX,XTXI,UP)
      RETURN
      END
```

```
                SUBROUTINE POINTS1(XCP,NPTS,PTS)

c       The  regression data sets are determined.
c       The system of interest is collapsed
c       into a pseudo-quartenary system
c       ( 3 component systems are perturbed from
c       their 'real' form). Each regression set
c       is determined by perturbing a pseudo-
c       component and then re-normalizing.
c       ****************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        INTEGER L,H,HIGH(10),LOW(10)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/INIT/XLIM(10,10),IV,IL
        COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     &  DGAMX(20,20),DGAMY(20,20)
        COMMON/REG/BPTS(50,10)
        REAL*8 XCP(20),PTS(50,10),DIFF1,DIFF2,XBB(10)
        COMMON/PERTA/PERT1,PERT2,PERT3,PERT4,PERT5
        REAL*8 KMAX,KMIN,S,SH,SL,SOL(10)
        REAL*8 XA,XB,XC,XD
c       ****************************************
        DATA XA,XB,XC,XD/0.0,0.0,0.0,0.0/
        DATA KMIN,KMAX,L,H/0.0,0.0,0,0/
        DO 5 I=1,N
                XBB(I)=0.0
                HIGH(I)=0.0
                LOW(I)=0.0
5       CONTINUE

c       ****************************************
        IF(N.NE.3)GOTO 15
        DO 10 I=1,N
                XC=XCP(I)
                IF(I.EQ.IV)XA=XCP(I)
                IF(I.EQ.IL)XB=XCP(I)
10      CONTINUE
        GOTO 45
c       ****************************************
c       For # components greater than or equal
c       to 4: One pseudo-component is the IV
c       solvent or key component. Another is the
c       IL component. The remaining two pseudo-
c       compositions are defined by the closeness
c       of their K-values to either K(IV) or K(IL).

15      DO 20 I=1,N
                SOL(I)=0.
```

```
                IF(I.EQ.IV)XA=XCP(I)
                IF(I.EQ.IL)XB=XCP(I)
                IF(I.NE.IV.AND.I.NE.IL)SOL(I)=RK(I)
20          CONTINUE
            KMAX=0.0
            DO 25 I=1,N
                IF(SOL(I).LE.KMAX)GOTO15
                KMAX=SOL(I)
                H=I
25          CONTINUE
              HIGH(H)=H
              KMIN=KMAX
            DO 30 I=1,N
                IF(SOL(I).GE.KMIN.OR.SOL(I).EQ.0.0)GOTO20
                KMIN=SOL(I)
                L=I
30          CONTINUE
              LOW(L)=L
            DO 35 I=1,N
                IF(I.EQ.IV.OR.I.EQ.IL)GOTO25
                IF(I.EQ.L)XC=XCP(I)
                IF(I.EQ.H)XD=XCP(I)
35          CONTINUE

            DO 40 I=1,N
              IF(I.EQ.IV.OR.I.EQ.IL.OR.I.EQ.H.OR.I.EQ.L)GOTO30
              DIFF1=DABS(RK(I)-KMIN)
              DIFF2=DABS(KMAX-RK(I))
               IF(DIFF1.LE.DIFF2)THEN
                    XC=XC+XCP(I)
                    LOW(I)=I
               ELSE
                    XD=XD+XCP(I)
                    HIGH(I)=I
              END IF
40          CONTINUE

45          XBB(1)=XA
            XBB(2)=XB
            XBB(3)=XC
            XBB(4)=XD

            DO 50 I=1,4
                BPTS(1,I)=XBB(I)
50          CONTINUE

c           ***********************************************
c           Each component of the original data set is
c           perturbed by multiplying it by a factor. The
```

```
c       procedure works as follows:
c         (1) Consider cmpt #1
c         (2) multiply it by a factor
c         (3) renormalize this perturbed cmpt 1
c             with the unperturbed cmpts 2,3 & 4.
c         (4) Consider this set # 2 (set # 1 is
c             the original set).
c         (5) Repeat with cmpt # 2 of set # 1 etc...

        DO 80 I=2,NPTS
                DO 55 J=1,4
                        BPTS(I,J)=XBB(J)
55              CONTINUE
                IF(N.NE.3)GOTO 60
        IF(I.GE.20.AND.I.LE.22)BPTS(I,I-19)=XBB(I-19)*PERT7
        IF(I.GE.17.AND.I.LE.19)BPTS(I,I-16)=XBB(I-16)*PERT6
        IF(I.GE.14.AND.I.LE.16)BPTS(I,I-13)=XBB(I-13)*PERT5
        IF(I.GE.11.AND.I.LE.13)BPTS(I,I-10)=XBB(I-10)*PERT4
        IF(I.GE.8.AND.I.LE.10)BPTS(I,I-7)=XBB(I-7)*PERT3
        IF(I.GE.5.AND.I.LE.7)BPTS(I,I-4)=XBB(I-4)*PERT2
        IF(I.LT.5)BPTS(I,I-1)=XBB(I-1)*PERT1
60              IF(N.EQ.3)GOTO 65
        IF(I.GE.26.AND.I.LE.29)BPTS(I,I-25)=XBB(I-25)*PERT7
        IF(I.GE.22.AND.I.LE.25)BPTS(I,I-21)=XBB(I-21)*PERT6
        IF(I.GE.18.AND.I.LE.21)BPTS(I,I-17)=XBB(I-17)*PERT5
        IF(I.GE.14.AND.I.LE.17)BPTS(I,I-13)=XBB(I-13)*PERT4
        IF(I.GE.10.AND.I.LE.13)BPTS(I,I-9)=XBB(I-9)*PERT3
        IF(I.GE.6.AND.I.LE.9)BPTS(I,I-5)=XBB(I-5)*PERT2
        IF(I.LT.6)BPTS(I,I-1)=XBB(I-1)*PERT1
65              S=0.0
                DO 70 J=1,4
                        IF(BPTS(I,J).LT.0.0)BPTS(I,J)=.001
                        S=S+BPTS(I,J)
70              CONTINUE
                DO 75 J=1,4
                        BPTS(I,J)=BPTS(I,J)/S
75              CONTINUE
80      CONTINUE
c       ***********************************************
c       Convert back into real compositions PTS.
c       If the system of interest has more than
c       4 components, the solute pseudo-components
c       are converted back to real compositions
c       according to the fraction of the solutes that
c       a component of interest represents.

        SL=1-XA-XB-XC
        SH=1-XA-XB-XD
        DO 90 I=1,NPTS
```

```
                DO 85 J=1,N
                        IF(J.EQ.IV.OR.J.EQ.IL)GOTO 85
                        PTS(I,J)=BPTS(I,3)
                        IF(N.EQ.3)GOTO 85
        IF(J.EQ.LOW(J))PTS(I,J)=BPTS(I,3)*XCP(J)/SH
        IF(J.EQ.HIGH(J))PTS(I,J)=BPTS(I,4)*XCP(J)/SL
85              CONTINUE
                PTS(I,IV)=BPTS(I,1)
                PTS(I,IL)=BPTS(I,2)
90      CONTINUE
        RETURN
        END
```

```
      SUBROUTINE INITLA (NUM,MODEL,KIND)

c     INITLA provides the data points
c     that will be used in the LA
c     parameter regression. This can
c     be used in conjunction with NRLA1
c     and POINTS
c     ****************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BASEACT/XBGAM(20,20),YBGAM(20,20)
      COMMON/BASE/XBSTAR(20,20),XCSTAR(20,20),XDSTAR(20,20)
      COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     & DGAMX(20,20),DGAMY(20,20)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/INIT/XLIM(10,10),IV,IL
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     & GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     & YPTS(50,10),DPTS(50,3)
      COMMON/UPDATE/INXTX,INXTY
      COMMON/RPOINTS/IDP
      INTEGER IPTS(50)
      REAL*8 XTX(21,21),XTXI(21,21),LPTS(50,3),VPTS(50,3)
      REAL*8 PERT1,PERT2,PERT3,PERT4,PERT5
      REAL*8 PERT6,PERT7,PERT8,PERT9,PERT10,PERT11
c     OPEN(FILE='EEX',UNIT=92,STATUS='UNKNOWN')
c     OPEN(FILE='EEY',UNIT=93,STATUS='UNKNOWN')
c     ****************************************
c     Number of data sets is determined.

      WRITE(6,*)'INPUT NDPTS'
      READ(6,*)NDPTS
c     ****************************************
c     Subroutine POINTS is called to set up
c     the regression data sets.
c     POINTS groups all solutes into
c     one pseudo-component. These are
c     converted back into real compositions
c     by multiplying the pseudo-component
c     by the fraction of the real solutes that
c     the component of interest represents.

      S=1.-(X(IV)+X(IL))
      CALL POINTS(X,NDPTS)

      DO 70 I=1,NDPTS
            DO 65 J=1,N
                  IF(J.EQ.IV.OR.J.EQ.IL)GOTO65
                  XPTS(I,J)=DPTS(I,1)*X(J)/S
```

```
65              CONTINUE
                XPTS(I,IV)=DPTS(I,2)
                XPTS(I,IL)=DPTS(I,3)
70      CONTINUE

        DO 90 I=1,NDPTS
                SX=0.
        DO 80 J=1,N
                IF(XPTS(I,J).LT.0.)XPTS(I,J)=.001
                SX=SX+XPTS(I,J)
80      CONTINUE
        DO 85 J=1,N
                XPTS(I,J)=XPTS(I,J)/SX
85      CONTINUE
90      CONTINUE

        S=1.-(Y(IV)+Y(IL))
        CALL POINTS(Y,NDPTS)
        DO 100 I=1,NDPTS
                DO 95 J=1,N
                        IF(J.EQ.IV.OR.J.EQ.IL)GOTO95
                        YPTS(I,J)=DPTS(I,1)*Y(J)/S
95              CONTINUE
                YPTS(I,IV)=DPTS(I,2)
                YPTS(I,IL)=DPTS(I,3)
100     CONTINUE
        DO 120 I=1,NDPTS
                SY=0.
        DO 110 J=1,N
                IF(YPTS(I,J).LT.0.)YPTS(I,J)=.001
                SY=SY+YPTS(I,J)
110     CONTINUE
        DO 115 J=1,N
                YPTS(I,J)=YPTS(I,J)/SY
115     CONTINUE
120     CONTINUE
C       ****************************************
125     DO 130 I=1,NDPTS
                IPTS(I)=I
130     CONTINUE

C       Get 'heavy' phase activity coefficients

        IF(MODEL.EQ.1)CALL NEWMOD(IPTS,1,NDPTS,NUM,XTX,XTXI)
        IF(MODEL.EQ.2)CALL NEWMOD(IPTS,1,NDPTS,NUM,XTX,XTXI)
        IF(MODEL.EQ.3)
     &  CALL NMEXPTER(IPTS,1,NDPTS,NUM,XTX,XTXI,0)
        IF(MODEL.EQ.4)
     &  CALL NMEXPQUAT(IPTS,1,NDPTS,NUM,XTX,XTXI,0)
```

```
c       Get 'light' phase activity coefficients

        IF(KIND.EQ.1.AND.MODEL.EQ.1)
     &          CALL NEWMOD(IPTS,2,NDPTS,NUM,XTX,XTXI)
        IF(KIND.EQ.1.AND.MODEL.EQ.2)
     &          CALL NEWMOD(IPTS,2,NDPTS,NUM,XTX,XTXI)
        IF(KIND.EQ.1.AND.MODEL.EQ.3)
     &          CALL NMEXPTER(IPTS,2,NDPTS,NUM,XTX,XTXI,0)
        XC=XCP(I)
        IF(KIND.EQ.1.AND.MODEL.EQ.4)
     &          CALL NMEXPQUAT(IPTS,2,NDPTS,NUM,XTX,XTXI,0)

        INXTX=NDPTS+1
        INXTY=NDPTS+1
        RETURN
        END
```

```
c                SUBROUTINE POINTS(XCP,NPTS)

c        The locations of the regression data
c        points are determined. This is used
c        with INITLA and NRLA1. A real mixture
c        is reduced to a ternary mixture.
c        Perturbations are performed on this
c        pseudo-ternary by geometric manipulations.
c        ****************************************

c        IMPLICIT REAL*8 (A-H,O-Z)
c        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
c        COMMON/INIT/XLIM(10,10),IV,IL
c        COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
c     &  GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
c    &   YPTS(50,10),DPTS(50,3)
c        REAL*8 XCP(20),CT(15,4)
c        ****************************************
c        Rectangular placement of data points in
c        various orientations is set up.
c        ****************************************
c        Option 1:
c        The following will set up 5,8,11,0R 20
c        equally spaced points on semi-circle of
c        radius RAD.

c        DO 1 I=1,4
c          .       CT(1,I)=0.
c1       CONTINUE
c        WRITE(6,*)'INPUT RADIUS'
c        READ(6,*)RAD
c        RAD=0.05
c        ANGLE=0.
c        DO 2 I=2,NPTS
c                ANGLE=ANGLE*.01745329
c                CT(I,1)=RAD*DCOS(ANGLE)
c                CT(I,2)=(-RAD)*DSIN(ANGLE)
c                CT(I,3)=RAD*DSIN(ANGLE)
c                CT(I,4)=RAD*DCOS(ANGLE)
c                ANGLE=ANGLE/.01745329
c                IF(NPTS.EQ.5)ANGLE=ANGLE+60.
c                IF(NPTS.EQ.8)ANGLE=ANGLE+30.
c                IF(NPTS.EQ.11)ANGLE=ANGLE+20.
c                IF(NPTS.EQ.20)ANGLE=ANGLE+10.
c2       CONTINUE

c        ****************************************
c        Option 2:
c        Scheme for six points on a pyramid.
```

```
c       DATA CT /0.0,0.05,-.05,0.025,-0.025,0.0,0.0,
c &     0.0,0.0,-0.0433,-.0433,-.0866,0.0,0.0,0.0,
c &     0.0433,0.0433,.0866,0.0,0.05,-0.05,0.025
c &     ,-0.025,0.0/

c       ****************************************
c       Option 3:
c       The following scheme will set up 4 equally
c       spaced points on two semi-circles of radii
c       .025,.05 (these may be varied)

c       DO 1 I=1,4
c               CT(1,I)=0.
c1      CONTINUE
c       ANGLE=0.
c       DO 2 I=2,5
c               ANGLE=ANGLE*.01745329
c               CT(I,1)=.05*DCOS(ANGLE)
c               CT(I,2)=(-.05)*DSIN(ANGLE)
c               CT(I,3)=.05*DSIN(ANGLE)
c               CT(I,4)=.05*DCOS(ANGLE)
c               ANGLE=ANGLE/.01745329
c               ANGLE=ANGLE+60.D00
c2      CONTINUE
c       ANGLE=0.0
c       DO 3 I=6,9
c               ANGLE=ANGLE*.01745329
c               CT(I,1)=.025*DCOS(ANGLE)
c               CT(I,2)=(-.025)*DSIN(ANGLE)
c               CT(I,3)=.025*DSIN(ANGLE)
c               CT(I,4)=.025*DCOS(ANGLE)
c               ANGLE=ANGLE/.01745329
c               ANGLE=ANGLE+60.D00
c3      CONTINUE
c       ****************************************
c       Option 4:
c       Points on a catenay.

c       RAD=.05
c       EXX=-.225
c       DO 1 I=1,4
c       CT(1,I)=0.0
c1      CONTINUE
c       DO 2 I=2,NPTS
c               EXX=EXX+.025
c               CT(I,1)=EXX
c               CT(I,2)=-RAD*DCOSH(EXX/RAD)
c               CT(I,3)=-CT(I,2)
```

```
c               CT(I,4)=EXX
c2        CONTINUE
c         ****************************************
c         Option 5:
c         Points on Witch of Agnesi

c         RAD=.025
c         EXX=-.1
c         DO 1 I=1,4
c                 CT(1,I)=0.0
c1        CONTINUE
c         DO 2 I=2,NPTS
c                 EXX=EXX+.0125
c                 CT(I,1)=EXX
c                 CT(I,2)=(8*RAD**3)/(EXX**2+4*RAD**2)
c                 CT(I,3)=-CT(I,2)
c                 CT(I,4)=EXX
c2        CONTINUE
c         ****************************************
c         Option 6:
c         Points on a rectangle.

c         DATA CT /0.0,.025,.05,-.025,0.0,0.0,-.05,.025,
c     & -.025,.025,-.025,-.05,.05,-.05,.05,0.0,0.0,0.0,
c     & 0.0,-.025,-.05,0.0,-.025,-.025,-.0433,-.0433,
c     &  -.025,-.025,-.05,-.05,0.0,0.0,0.0,0.0,.025,.05,
c     &  0.0,.025,.025,.0433,.0433,.025,.025,.05,.05,0.0,
c     &  .025,.05,-.025,0.0,0.0,-.05,.025,.025,.025,-.025,
c     &  -.05,.05,-.05,.05/
c         ****************************************
c         Combine the real incoming compositions
c         into pseudo-compositions XA,XB,XC. This
c         base point can then be manipulated in
c         ternary space.
c
c         XA=0.
c         DO 5 I=1,N
c                 IF(I.EQ.IV)XB=XCP(I)
c                 IF(I.EQ.IL)XC=XCP(I)
c                 IF(I.NE.IV.AND.I.NE.IL)XA=XA+XCP(I)
c5        CONTINUE
c         ****************************************
c         Find the rectangular coordinates of the
c         base point.

c         X0=1.-(XB+XA/2.)
c         BETA=1.-X0
c         Y0=XA*.866
c         ****************************************
```

```
c       Calculate the slope of a line drawn from
c       the pure minority solvent to the base point.

c       IF(XB.GE.XC)RM0=-Y0/BETA
c       IF(XB.LT.XC)RM0=Y0/X0
c       ****************************************
c       Find the angle of rotation

c       ALF=DATAN(-1./RM0)
c       ****************************************
c       The translation-rotation equations are set
c       up. The compositions are then transformed
c       back to triangular coordinates.

c       CA=DCOS(ALF)
c       SA=DSIN(ALF)
c       DO 10 I=1,NPTS

c       dpts below represent XA,XB,XC respectively

c       DPTS(I,1)=(Y0+CT(I,3)*CA+CT(I,4)*SA)/.866
c       DPTS(I,2)=1.-(.5*DPTS(I,1))-(X0+CT(I,1)*CA+CT(I,2)*SA)
c       DPTS(I,3)=1.-(DPTS(I,1)+DPTS(I,2))
c10     CONTINUE
c       RETURN
c       END
```

```
      SUBROUTINE NEWMOD(IPTS,IPHS,NPTS,NUM,XTX,XTXI)

c     NEWMOD will solve the regression equations for
c     the LA model (model#1 or model#2) parameters.
c     ****************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     & GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     & YPTS(50,10),DPTS(50,3)
      REAL*8  DUML(20),DUML1(20),DUML2(20,20),XABC(50,10),
     & XTY(20)
      REAL*8 XTX(NUM,NUM),XTXI(NUM,NUM),WKAREA(600)
      REAL*8 LHS(50),QRAUX(10),RSD(50),B(10),U(10),W(10)
      REAL*8 S(10),E(10),UU(30,30),V(10,10),DUM1(30)
      REAL*8 YBEST(50,10),XBEST(50,10),DUM(10)
      INTEGER RLEAP,LDX,IPTS(50)
c     ****************************************
c     LINPACK library routines utilize the
c     following variables.

c     LDX=50
c     LDU=30
c     LDV=10
c     ****************************************
c     If RLEAP = 1, then use an IMSL routine
c     that will obtain a best regression model.

      RLEAP=0
c     ****************************************

      IFDER=0

c     Obtain the activity coefficients of the
c     regression sets, and set up the left hand
c     sides of the set of simultaneous
c     equations that will be solved by
c     least-squares regression.

      DO 25 I=1,NPTS
            IND=IPTS(I)
c           IF(IPHS.EQ.1.AND.GXPTS(IND,1).NE.0.)GOTO25
c           IF(IPHS.EQ.2.AND.GYPTS(IND,1).NE.0.)GOTO25
                  DO 5 J=1,N
                  IF(IPHS.EQ.1)DUML(J)=XPTS(IND,J)
                  IF(IPHS.EQ.2)DUML(J)=YPTS(IND,J)
5                 CONTINUE
            SD=0.
```

```
                        DO 10 J=1,N
                                IF(DUML(J).LT.0.)DUML(J)=0.
                                SD=SD+DUML(J)
10                      CONTINUE
                        DO 15 J=1,N
                                DUML(J)=DUML(J)/SD
15                      CONTINUE
                CALL NRTL(DUML,DUML1,DUML2)
                DO 20 J=1,N
                IF(IPHS.EQ.1)GXPTS(IND,J)=1.987*T*DLOG(DUML1(J))
                IF(IPHS.EQ.2)GYPTS(IND,J)=1.987*T*DLOG(DUML1(J))
20              CONTINUE
25      CONTINUE
c       ****************************************
c       Set up the pseudo-component mole
c       fractions.

        DO 85 I=1,N
                DO 45 J=1,NPTS
                IND=IPTS(J)
c       ****************************************
c       Pseudo-comps. of the ternary LA are set
c       up.

                IF(NUM.NE.3)GOTO 35
                        XB=0.
                        XC=0.
                        DO 30 K=1,N
        IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.1)XB=XB+XPTS(IND,K)
        IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.-1)XC=XC+XPTS(IND,K)
        IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.1)XB=XB+YPTS(IND,K)
        IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.-1)XC=XC+YPTS(IND,K)
30                      CONTINUE
                XABC(J,1)=XB**2
                XABC(J,2)=XC**2
                XABC(J,3)=XB*XC
c       ****************************************
c       Pseudo-comps. of the quartenary model
c       are set up.

35              IF(NUM.NE.6)GOTO45

                        XB=0.
                        XC=0.
                        XD=0.
                        DO 40 K=1,N
        IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.1)XB=XB+XPTS(IND,K)
        IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.2)XC=XC+XPTS(IND,K)
        IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.3)XD=XD+XPTS(IND,K)
```

```
      IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.1)XB=XB+YPTS(IND,K)
      IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.2)XC=XC+YPTS(IND,K)
      IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.3)XD=XD+YPTS(IND,K)
40                  CONTINUE
            XABC(J,1)=XB**2
            XABC(J,2)=XC**2
            XABC(J,3)=XD**2
            XABC(J,4)=XB*XC
            XABC(J,5)=XB*XD
            XABC(J,6)=XC*XD
c     ****************************************
45          CONTINUE
c     ****************************************
c     Option to examine best regression model
c     Set up dummy variables for gxpts and xabc.
c     (which are y & x  in  y = x b)

c           DO 41 J=1,NPTS
c                   YBEST(J,I)=GXPTS(J,I)
c           DO 42 K=1,NUM
c                   XBEST(J,K)=XABC(J,K)
c42         CONTINUE
c41         CONTINUE
c      IF(RLEAP.EQ.1)CALL BESTR(XBEST,YBEST,NPTS,NUM,I)
c     ****************************************
c     Begin the actual regression.

            DO 60 J=1,NUM
                  DO 55 K=1,NUM
                   SUM=0.
                        DO 50 L=1,NPTS
                              SUM=SUM+XABC(L,J)*XABC(L,K)
50                      CONTINUE
                        XTX(J,K)=SUM
55                CONTINUE
60          CONTINUE
c     ****************************************
c     Call Library subroutine LINV2F or MATINV.

c           CALL MATINV(XTX,NUM,NUM)
            CALL LINV2F(XTX,NUM,NUM,XTXI,-1,WKAREA,IER)
            WRITE(6,*)'IER FROM NEWMOD',IER
            DO 70 J=1,NUM
                  SUM=0.
                  DO 65 K=1,NPTS
                  IND=IPTS(K)
            IF(IPHS.EQ.1)SUM=SUM+XABC(K,J)*GXPTS(IND,I)
            IF(IPHS.EQ.2)SUM=SUM+XABC(K,J)*GYPTS(IND,I)
65                CONTINUE
```

```
                        XTY(J)=SUM
70              CONTINUE
c       ***********************************
c       A number of LINPACK library routines
c       can be used to analyze the regression
c       by singular value decomposition.

c       DO 61 J=1,NPTS
c               IF(IPHS.EQ.1)LHS(J)=GXPTS(J,I)
c               IF(IPHS.EQ.2)LHS(J)=GYPTS(J,I)
c61     CONTINUE
c       CALL DSVDC(XBEST,LDX,NPTS,NUM,S,E,UU,LDU,V,LDV,
c&         DUM1,00,INFO)
c
c       DO 63 J=1,NUM
c               WRITE(6,*)'S',S(J)
c63     CONTINUE
c
c       CALL DQRDC(XABC,LDX,NPTS,NUM,QRAUX,JDUM,DUM,0)
c       CALL DQRSL(XABC,LDX,NPTS,NUM,QRAUX,
c&      LHS,DUM,RSD,B,RSD,DUM,110,INFO)
c       ***********************************
                DO 80 J=1,NUM
                        SUM=0.
                                DO 75 K=1,NUM
                                SUM=SUM+XTXI(J,K)*XTY(K)
75                              CONTINUE
                        IF(IPHS.EQ.1)PARMX(I,J)=SUM
                        IF(IPHS.EQ.2)PARMY(I,J)=SUM
80              CONTINUE
85      CONTINUE
        RETURN
        END
```

```
c       SUBROUTINE BESTR(XXX,YYY,NPTS,NUM,I)
c       ************************************
c       Obtain the corrected sums of squares
c       and cross product matrix, and find
c       a number of best regression subsets
c       from a full regression model
c       ************************************
c       IMPLICIT REAL*8 (A-H,O-Z)
c       INTEGER IX,NBR(6),IER,NPTS,NUM
c       INTEGER KZ,IJOB(4),IXS(10),IXV(10)
c       INTEGER NVAR(90),IXB(10),IB,IW(450)
c       DIMENSION XXX(50,10),YYY(50,10),TEMP(10)
c       DIMENSION XM(10),VCV(60)
c       DIMENSION RR(700),STAT(20),BEST(50,4)
c       DIMENSION WK(100)
c       ************************************
c       DO 5 J=1,NPTS
c              XXX(J,NUM+1)=YYY(J,I)
c5      CONTINUE
c       NBR(1)=NUM+1
c       NBR(2)=NPTS
c       NBR(3)=NPTS
c       NBR(4)=1
c       NBR(5)=1
c       NBR(6)=1
c       IX=50
c       CALL BECOVM(XXX,IX,NBR,TEMP,XM,VCV,IER)
c       WRITE(6,*)'IER FROM BECOMV',IER
c       KZ=NUM+1
c       IJOB(1)=NPTS-1
c       IJOB(2)=1
c       IJOB(3)=1
c       IJOB(4)=2
c       IB=50
c       CALL RLEAP(VCV,KZ,IJOB,IXS,STAT,IXV,NVAR,IXB,
c    &             BEST,IB,WK,IW,IER)
c       WRITE(6,*)'FROM RLEAP IJOB IS',IJOB(3)
c       WRITE(6,*)'IER FROM RLEAP IS',IER
c       DO 11 J=1,50
c       WRITE(6,*)'IW',IW(J)
c11     CONTINUE
c       DO 12 J=1,20
c       DO 13 K=1,4
c       WRITE(6,*)'BEST J,K',J,K,BEST(J,K)
c13     CONTINUE
c12     CONTINUE
c       RETURN
c       END
        SUBROUTINE NMEXPTER(IPTS,IPHS,NPTS,NUM,XTX,XTXI,MB)
```

```
c       NMEXPTER regresses for the parameters of
c       a Taylor series expansion of the pseudo-
c       ternary 2-suffix Margules equation.
c       ***************************************

        IMPLICIT REAL*8 (A-H,O-Z)
        COMMON/BASEACT/XBGAM(20,20),YBGAM(20,20)
        COMMON/BASE/XBSTAR(20,20),XCSTAR(20,20),XDSTAR(20,20)
        COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     &  DGAMX(20,20),DGAMY(20,20)
        COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
        COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     &  GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     &  YPTS(50,10),DPTS(50,3)
        REAL*8  DUML(20),DUML1(20),DUML2(20,20),XABC(50,10),
     &  XTY(20)
        REAL*8 XTX(NUM,NUM),XTXI(NUM,NUM),WKAREA(600)
        REAL*8 XN(20),YN(20)
        REAL*8 GX(30,10),GY(30,10)
        INTEGER IPTS(50)
c       ****************************************
        IFDER=0
c       ****************************************
c       Normalize phase compositions.

        SX=0.
        SY=0.
        DO 15 I=1,N
                IF(IPHS.EQ.1)XN(I)=X(I)
                IF(IPHS.EQ.2)YN(I)=Y(I)
                IF(IPHS.EQ.1.AND.XN(I).LT.0.)XN(I)=0.
                IF(IPHS.EQ.2.AND.YN(I).LT.0.)YN(I)=0.
                IF(IPHS.EQ.1)SX=SX+XN(I)
                IF(IPHS.EQ.2)SY=SY+YN(I)
15      CONTINUE
        DO 21 I=1,N
                IF(IPHS.EQ.1)XN(I)=XN(I)/SX
                IF(IPHS.EQ.2)YN(I)=YN(I)/SY
21      CONTINUE
c       ****************************************
c       The base point (XBSTAR,XCSTAR) is
c       created

        DO 25 I=1,N
                XB=0.
                XC=0.
                DO 20 K=1,N
        IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.1)XB=XB+XN(K)
```

```
      IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.-1)XC=XC+XN(K)
      IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.1)XB=XB+YN(K)
      IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.-1)XC=XC+YN(K)
20          CONTINUE
            XBSTAR(I,IPHS)=XB
            XCSTAR(I,IPHS)=XC
25    CONTINUE
c     ****************************************
c     If the initial model is being created,
c     obtain activity coefficients for each
c     of the data sets, and set up the left
c     hand sides of the simultaneous equations
c     whose parameteres will be solved for
c     by least-squares regression.

      IF(MB.EQ.1)GOTO 52
      DO 5 I=1,N
         IF(IPHS.EQ.1)XBGAM(I,1)=GAMX(I)
         IF(IPHS.EQ.2)YBGAM(I,2)=GAMY(I)
5     CONTINUE
      DO 50 I=1,NPTS
            IND=IPTS(I)
c           IF(IPHS.EQ.1.AND.GXPTS(IND,1).NE.0.)GOTO50
c           IF(IPHS.EQ.2.AND.GYPTS(IND,1).NE.0.)GOTO50
                  DO 30 J=1,N
                  IF(IPHS.EQ.1)DUML(J)=XPTS(IND,J)
                  IF(IPHS.EQ.2)DUML(J)=YPTS(IND,J)
30                CONTINUE
            SD=0.
                  DO 35 J=1,N
                        IF(DUML(J).LT.0.)DUML(J)=0.
                        SD=SD+DUML(J)
35                CONTINUE
                  DO 40 J=1,N
                        DUML(J)=DUML(J)/SD
40                CONTINUE
            CALL NRTL(DUML,DUML1,DUML2)
            DO 43 J=1,N
               IF(IPHS.EQ.1)GX(I,J)=DUML1(J)
               IF(IPHS.EQ.2)GY(I,J)=DUML1(J)
43          CONTINUE
            DO 45 J=1,N
      IF(IPHS.EQ.1)GXPTS(IND,J)=(1.987*T*DLOG(DUML1(J)))
     &-(1.987*T*DLOG(XBGAM(J,1)))
      IF(IPHS.EQ.2)GYPTS(IND,J)=(1.987*T*DLOG(DUML1(J)))
     &-(1.987*T*DLOG(YBGAM(J,2)))
45          CONTINUE
50    CONTINUE
c     ****************************************
```

```
c       If the model is being updated, replace the
c       old base point activity coefficients with
c       new NRTL values.

52      IF(MB.EQ.0)GOTO53
        IF(IPHS.EQ.1)CALL NRTL(XN,DUML1,DUML2)
        IF(IPHS.EQ.2)CALL NRTL(YN,DUML1,DUML2)
        DO 6 I=1,N
           IF(IPHS.EQ.1)XBGAM(I,1)=DUML1(I)
           IF(IPHS.EQ.2)YBGAM(I,2)=DUML1(I)
6       CONTINUE
        DO 46 I=1,NPTS
                DO 51 J=1,N
        IF(IPHS.EQ.1)GXPTS(I,J)=(1.987*T*DLOG(GX(I,J)))
     &  -(1.987*T*DLOG(XBGAM(J,1)))
        IF(IPHS.EQ.2)GYPTS(I,J)=(1.987*T*DLOG(GY(I,J)))
     &  -(1.987*T*DLOG(YBGAM(J,2)))
51              CONTINUE
46      CONTINUE

c       ******************************************
c       Set up the pseudo-compositions and solve
c       for the parameters of the N equations
c       by least-squares regression.

53      DO 100 I=1,N
                DO 60 J=1,NPTS
                        IND=IPTS(J)
                        XB=0.
                        XC=0.
                        DO 55 K=1,N
        IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.1)XB=XB+XPTS(IND,K)
        IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.-1)XC=XC+XPTS(IND,K)
        IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.1)XB=XB+YPTS(IND,K)
        IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.-1)XC=XC+YPTS(IND,K)
55                      CONTINUE
                XABC(J,1)=(XB-XBSTAR(I,IPHS))
                XABC(J,2)=(XC-XCSTAR(I,IPHS))
                XABC(J,3)=(XB-XBSTAR(I,IPHS))**2
                XABC(J,4)=(XC-XCSTAR(I,IPHS))**2
        XABC(J,5)=(XB-XBSTAR(I,IPHS))*(XC-XCSTAR(I,IPHS))
60              CONTINUE
                DO 75 J=1,NUM
                        DO 70 K=1,NUM
                        SUM=0.
                                DO 65 L=1,NPTS
                                SUM=SUM+XABC(L,J)*XABC(L,K)
65                              CONTINUE
                        XTX(J,K)=SUM
```

```
70                      CONTINUE
75                CONTINUE
c                 CALL MATINV(XTX,NUM,NUM)
                  CALL LINV2F(XTX,NUM,NUM,XTXI,-1,WKAREA,IER)
                  DO 85 J=1,NUM
                  SUM=0.
                          DO 80 K=1,NPTS
                                  IND=IPTS(K)
                  IF(IPHS.EQ.1)SUM=SUM+XABC(K,J)*GXPTS(IND,I)
                  IF(IPHS.EQ.2)SUM=SUM+XABC(K,J)*GYPTS(IND,I)
80                        CONTINUE
                  XTY(J)=SUM
85                CONTINUE
                  DO 95 J=1,NUM
                  SUM=0.
                          DO 90 K=1,NUM
                                  SUM=SUM+XTXI(J,K)*XTY(K)
90                        CONTINUE
                  IF(IPHS.EQ.1)PARMX(I,J)=SUM
                  IF(IPHS.EQ.2)PARMY(I,J)=SUM
95                CONTINUE
100       CONTINUE
          RETURN
          END
```

```
      SUBROUTINE NMEXP(IPTS,IPHS,NPTS,NUM,XTX,XTXI,UPDA)

c     NMEXP will create or update the parameters
c     of the 'true' linearized Taylor expansion
c     of the 2-suffix Margules equation. The model
c     is set up to accommodate up to six components.
c     ****************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BASEACT/XBGAM(20,20),YBGAM(20,20)
      COMMON/BASE1/X1S(10,10),X2S(10,10)
     &  ,X3S(10,10),X4S(10,10),
     &  X5S(10,10),X6S(10,10)
      COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     &  DGAMX(20,20),DGAMY(20,20)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     &  GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     &  YPTS(50,10),DPTS(50,3)
      COMMON/DLHS/DGXPTS(50,10),DGYPTS(50,10)
      REAL*8  XABC(50,10),XTY(20)
      REAL*8 XTX(NUM,NUM),XTXI(NUM,NUM),WKAREA(600)
      REAL*8 XN(20),YN(20),DUML1(20),DUML2(20,20)
      INTEGER IPTS(50),UPDA
c     ****************************************
      IFDER=0
c     ****************************************
c     Normalize the most recent values of the
c     phase compositions.

      SX=0.
      SY=0.
      DO 15 I=1,N
              IF(IPHS.EQ.1)XN(I)=X(I)
              IF(IPHS.EQ.2)YN(I)=Y(I)
              IF(IPHS.EQ.1.AND.XN(I).LT.0.)XN(I)=0.
              IF(IPHS.EQ.2.AND.YN(I).LT.0.)YN(I)=0.
              IF(IPHS.EQ.1)SX=SX+XN(I)
              IF(IPHS.EQ.2)SY=SY+YN(I)
15    CONTINUE
      DO 21 I=1,N
              IF(IPHS.EQ.1)XN(I)=XN(I)/SX
              IF(IPHS.EQ.2)YN(I)=YN(I)/SY
21    CONTINUE

c     ****************************************
c     Equate the base point to the most recent
c     values of the phase comps. and activity
c     coefficients.
```

```
      DO 22 I=1,N
               X1S(I,1)=XN(1)
               X2S(I,1)=XN(2)
               X3S(I,1)=XN(3)
               X4S(I,1)=XN(4)
               X5S(I,1)=XN(5)
               X6S(I,1)=XN(6)
      IF(IPHS.EQ.1)GOTO 22
               X1S(I,2)=YN(1)
               X2S(I,2)=YN(2)
               X3S(I,2)=YN(3)
               X4S(I,2)=YN(4)
               X5S(I,2)=YN(5)
               X6S(I,2)=YN(6)
22    CONTINUE
      DO 5 I=1,N
         IF(IPHS.EQ.1)XBGAM(I,1)=GAMX(I)
         IF(IPHS.EQ.2)YBGAM(I,2)=GAMY(I)
5     CONTINUE

c     ****************************************
c     If the model is to be updated, use the
c     most recent values of the xi and yi to
c     obtain activity coefficients from the
c     NRTL equation.

      IF(UPDA.EQ.0)GOTO 53
      IF(IPHS.EQ.1)CALL NRTL(XN,DUML1,DUML2)
      IF(IPHS.EQ.2)CALL NRTL(YN,DUML1,DUML2)
      DO 6 I=1,N
         IF(IPHS.EQ.1)XBGAM(I,1)=DUML1(I)
         IF(IPHS.EQ.2)YBGAM(I,2)=DUML1(I)
6     CONTINUE


53    DO 50 I=1,NPTS
               IND=IPTS(I)
               DO 45 J=1,N
               IF(IPHS.EQ.1)GXPTS(IND,J)=DGXPTS(IND,J)
     &         -(1.987*T*DLOG(XBGAM(J,1)))
               IF(IPHS.EQ.2)GYPTS(IND,J)=DGYPTS(IND,J)
     &         -(1.987*T*DLOG(YBGAM(J,2)))
45             CONTINUE
50    CONTINUE

c     ****************************************
c     Calculate the parameters of the LA by
c     least-squares regression.
```

```
      DO 100 I=1,N
             DO 60 J=1,NPTS
             IND=IPTS(J)
                    X1=XPTS(IND,1)
                    X2=XPTS(IND,2)
                    X3=XPTS(IND,3)
                    X4=XPTS(IND,4)
                    X5=XPTS(IND,5)
                    X6=XPTS(IND,6)
                    IF(IPHS.EQ.1)GOTO 61
                    X1=YPTS(IND,1)
                    X2=YPTS(IND,2)
                    X3=YPTS(IND,3)
                    X4=YPTS(IND,4)
                    X5=YPTS(IND,5)
                    X6=YPTS(IND,6)

61           XABC(J,1)=(X1-X1S(I,IPHS))
             XABC(J,2)=(X2-X2S(I,IPHS))
             XABC(J,3)=(X3-X3S(I,IPHS))
             XABC(J,4)=(X4-X4S(I,IPHS))
             XABC(J,5)=(X5-X5S(I,IPHS))
             XABC(J,6)=(X6-X6S(I,IPHS))
60           CONTINUE

             DO 75 J=1,NUM
                    DO 70 K=1,NUM
                    SUM=0.
                           DO 65 L=1,NPTS
                           SUM=SUM+XABC(L,J)*XABC(L,K)
65                         CONTINUE
                    XTX(J,K)=SUM
70                  CONTINUE
75           CONTINUE
             CALL LINV2F(XTX,NUM,NUM,XTXI,-1,WKAREA,IER)
             DO 85 J=1,NUM
             SUM=0.
                    DO 80 K=1,NPTS
                    IND=IPTS(K)
             IF(IPHS.EQ.1)SUM=SUM+XABC(K,J)*GXPTS(IND,I)
             IF(IPHS.EQ.2)SUM=SUM+XABC(K,J)*GYPTS(IND,I)
80                  CONTINUE
             XTY(J)=SUM
85           CONTINUE
             DO 95 J=1,NUM
             SUM=0.
                    DO 90 K=1,NUM
                           SUM=SUM+XTXI(J,K)*XTY(K)
```

```
90                    CONTINUE
                IF(IPHS.EQ.1)PARMX(I,J)=SUM
                IF(IPHS.EQ.2)PARMY(I,J)=SUM
95              CONTINUE
100     CONTINUE
        RETURN
        END
```

```
      SUBROUTINE NMEXPQUAT(IPTS,IPHS,NPTS,NUM,XTX,XTXI,MB)

c     NMEXPQUAT regresses for the parameters of
c     a Taylor series expansion of the pseudo-
c     quartenary 2-suffix Margules equation.
c     ****************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BASEACT/XBGAM(20,20),YBGAM(20,20)
      COMMON/BASE/XBSTAR(20,20),XCSTAR(20,20),XDSTAR(20,20)
      COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     & DGAMX(20,20),DGAMY(20,20)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     & GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     & YPTS(50,10),DPTS(50,3)
      REAL*8  DUML(20),DUML1(20),DUML2(20,20),XABC(50,10),
     & XTY(20)
      REAL*8 XTX(NUM,NUM),XTXI(NUM,NUM),WKAREA(600)
      REAL*8 XN(20),YN(20)
      INTEGER IPTS(50)
      REAL*8 GX(30,10),GY(30,10)
c     ****************************************
      IFDER=0
c     ****************************************
c     Normalize phase compositions.

      SX=0.
      SY=0.
      DO 10 I=1,N
            XN(I)=X(I)
            YN(I)=Y(I)
            IF(XN(I).LT.0.)XN(I)=0.
            IF(YN(I).LT.0.)YN(I)=0.
            SX=SX+XN(I)
            SY=SY+YN(I)
10    CONTINUE
      DO 15 I=1,N
            XN(I)=XN(I)/SX
            YN(I)=YN(I)/SY
15    CONTINUE
c     ****************************************
c     The base point (XBSTAR,XCSTAR,XDSTAR) is
c     created.

      DO 25 I=1,N
            XB=0.
            XC=0.
            XD=0.
```

```
                DO 20 K=1,N
                  IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.1)XB=XB+XN(K)
                  IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.2)XC=XC+XN(K)
                  IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.3)XD=XD+XN(K)
                  IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.1)XB=XB+YN(K)
                  IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.2)XC=XC+YN(K)
                  IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.3)XD=XD+YN(K)
20              CONTINUE
                XBSTAR(I,IPHS)=XB
                XCSTAR(I,IPHS)=XC
                XDSTAR(I,IPHS)=XD
25      CONTINUE

c       ****************************************
c       If the initial model is being created,
c       obtain activity coefficients for each
c       of the data sets, and set up the left
c       hand sides of the simultaneous equations
c       whose parameteres will be solved for
c       by least-squares regression.

        IF(MB.EQ.1)GOTO 52
        DO 5 I=1,N
           IF(IPHS.EQ.1)XBGAM(I,1)=GAMX(I)
           IF(IPHS.EQ.2)YBGAM(I,2)=GAMY(I)
5       CONTINUE
        DO 50 I=1,NPTS
                IND=IPTS(I)
c               IF(IPHS.EQ.1.AND.GXPTS(IND,1).NE.0.)GOTO50
c               IF(IPHS.EQ.2.AND.GYPTS(IND,1).NE.0.)GOTO50
                        DO 30 J=1,N
                          IF(IPHS.EQ.1)DUML(J)=XPTS(IND,J)
                          IF(IPHS.EQ.2)DUML(J)=YPTS(IND,J)
30                      CONTINUE
                SD=0.
                        DO 35 J=1,N
                          IF(DUML(J).LT.0.)DUML(J)=0.
                          SD=SD+DUML(J)
35                      CONTINUE
                        DO 40 J=1,N
                          DUML(J)=DUML(J)/SD
40                      CONTINUE

                CALL NRTL(DUML,DUML1,DUML2)
                DO 43 J=1,N
                   IF(IPHS.EQ.1)GX(I,J)=DUML1(J)
                   IF(IPHS.EQ.2)GY(I,J)=DUML1(J)
43              CONTINUE
                DO 45 J=1,N
```

```
      IF(IPHS.EQ.1)GXPTS(IND,J)=(1.987*T*DLOG(DUML1(J)))
     &  -(1.987*T*DLOG(XBGAM(J,1)))
      IF(IPHS.EQ.2)GYPTS(IND,J)=(1.987*T*DLOG(DUML1(J)))
     &  -(1.987*T*DLOG(YBGAM(J,2)))
45              CONTINUE
50    CONTINUE

c     ******************************************
c     If the model is being updated, replace the
c     old base point activity coefficients with
c     new NRTL values.

52    IF(MB.EQ.0)GOTO 53
      IF(IPHS.EQ.1)CALL NRTL(XN,DUML1,DUML2)
      IF(IPHS.EQ.2)CALL NRTL(YN,DUML1,DUML2)
      DO 6 I=1,N
         IF(IPHS.EQ.1)XBGAM(I,1)=DUML1(I)
         IF(IPHS.EQ.2)YBGAM(I,2)=DUML1(I)
6     CONTINUE
      DO 46 I=1,NPTS
              DO 51 J=1,N
      IF(IPHS.EQ.1)GXPTS(I,J)=(1.987*T*DLOG(GX(I,J)))
     &  -(1.987*T*DLOG(XBGAM(J,1)))
      IF(IPHS.EQ.2)GYPTS(I,J)=(1.987*T*DLOG(GY(I,J)))
     &  -(1.987*T*DLOG(YBGAM(J,2)))
51              CONTINUE
46    CONTINUE
c     ******************************************
c     Set up the pseudo-compositions and solve
c     for the parameters of the N equations
c     by least-squares regression.

53    DO 100 I=1,N
              DO 60 J=1,NPTS
                      IND=IPTS(J)
                      XB=0.
                      XC=0.
                      XD=0.
                      DO 55 K=1,N
      IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.1)XB=XB+XPTS(IND,K)
      IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.2)XC=XC+XPTS(IND,K)
      IF(IPHS.EQ.1.AND.IPSX(I,K).EQ.3)XD=XD+XPTS(IND,K)
      IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.1)XB=XB+YPTS(IND,K)
      IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.2)XC=XC+YPTS(IND,K)
      IF(IPHS.EQ.2.AND.IPSY(I,K).EQ.3)XD=XD+YPTS(IND,K)
55                    CONTINUE

              XABC(J,1)=(XB-XBSTAR(I,IPHS))
              XABC(J,2)=(XC-XCSTAR(I,IPHS))
```

```
                XABC(J,3)=(XD-XDSTAR(I,IPHS))
                XABC(J,4)=(XB-XBSTAR(I,IPHS))**2
                XABC(J,5)=(XC-XCSTAR(I,IPHS))**2
                XABC(J,6)=(XD-XDSTAR(I,IPHS))**2
          XABC(J,7)=(XB-XBSTAR(I,IPHS))*(XC-XCSTAR(I,IPHS))
          XABC(J,8)=(XB-XBSTAR(I,IPHS))*(XD-XDSTAR(I,IPHS))
          XABC(J,9)=(XC-XCSTAR(I,IPHS))*(XD-XDSTAR(I,IPHS))
60              CONTINUE
                DO 75 J=1,NUM
                        DO 70 K=1,NUM
                         SUM=0.
                                DO 65 L=1,NPTS
                                SUM=SUM+XABC(L,J)*XABC(L,K)
65                              CONTINUE
                                XTX(J,K)=SUM
70                      CONTINUE
75              CONTINUE
C               CALL MATINV(XTX,NUM,NUM)
                CALL LINV2F(XTX,NUM,NUM,XTXI,-1,WKAREA,IER)
                DO 85 J=1,NUM
                   SUM=0.
                        DO 80 K=1,NPTS
                         IND=IPTS(K)
                IF(IPHS.EQ.1)SUM=SUM+XABC(K,J)*GXPTS(IND,I)
                IF(IPHS.EQ.2)SUM=SUM+XABC(K,J)*GYPTS(IND,I)
80                      CONTINUE
                 XTY(J)=SUM
85              CONTINUE
                DO 95 J=1,NUM
                        SUM=0.
                                DO 90 K=1,NUM
                                SUM=SUM+XTXI(J,K)*XTY(K)
90                              CONTINUE
                        IF(IPHS.EQ.1)PARMX(I,J)=SUM
                        IF(IPHS.EQ.2)PARMY(I,J)=SUM
95              CONTINUE
100     CONTINUE
        RETURN
        END
```

```
      SUBROUTINE LATERN(X,GAM,DGAM,IPS,LCNVD)

c     The pseudo-ternary two-suffix Margules
c     is used as an activity coefficient correlation.
c     ***************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/OPT/NSS,KIND,NMT,NLA1
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/INIT/XLIM(10,10),IV,IL
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     & GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     & YPTS(50,10),DPTS(50,3)
      COMMON/UPDATE/INXTX,INXTY
      INTEGER IPTS(50),IPS(10,10)
      REAL*8 IB,IC,ID
      REAL*8 X(20),GAM(20),DGAM(20,20),DLNG(20)
c     ***************************************
      IFDER=0
      DATA R/1.987/
      IPHS=1
      IF(X(IV).GT.X(IL))IPHS=2
      IF(LCNVD.NE.1)GOTO 4
      CALL NRTL(X,DLNG,DGAM)
c     ***************************************
c     After the LA model has converged to
c     its solution, activity coefficients
c     are then provided by the NRTL. These
c     additional phase compositions and
c     activity coefficients can be added
c     to the regression data. The
c     model can then be updated.

c     IF(IPHS.EQ.1.AND.INXTX.GT.66)GOTO4
c     IF(IPHS.EQ.2.AND.INXTY.GT.66)GOTO4
c     DO 1 I=1,N
c             IF(IPHS.EQ.1)XPTS(INXTX,I)=X(I)
c             IF(IPHS.EQ.2)YPTS(INXTY,I)=X(I)
c     IF(IPHS.EQ.1)GXPTS(INXTX,I)=R*T*DLOG(DLNG(I))
c     IF(IPHS.EQ.2)GYPTS(INXTY,I)=R*T*DLOG(DLNG(I))
c1    CONTINUE
c     IF(IPHS.EQ.1)NPTS=INXTX
c     IF(IPHS.EQ.2)NPTS=INXTY
c     IF(IPHS.EQ.1)INXTX=INXTX+1
c     IF(IPHS.EQ.2)INXTY=INXTY+1

4     DO 20 I=1,N
             XB=0.
             XC=0.
```

```
                DO 5 J=1,N
                        IB=0.
                        IC=0.
                        IF(IPS(I,J).EQ.1)IB=1.
                        IF(IPS(I,J).EQ.-1)IC=1.
                        XB=XB+X(J)*IB
                        XC=XC+X(J)*IC
5               CONTINUE
        C1=PARMX(I,1)
        C2=PARMX(I,2)
        C3=PARMX(I,3)
        IF(IPHS.EQ.1)GOTO10
        C1=PARMY(I,1)
        C2=PARMY(I,2)
        C3=PARMY(I,3)
c       ************************************
c       The partial derivatives of the activity
c       coefficient with respect to xi,xb,xc
c       (DG1,DG2,DG3) are calculated.

10      GAM(I)=DEXP( ((C1*XB**2)+(C2*XC**2)+(C3*XB*XC))/(R*T))
        DG1=-2.*GAM(I)*DLOG(GAM(I))
        DG2=DG1+GAM(I)*(2.*C1*XB+C3*XC)/(R*T)
        DG3=DG1+GAM(I)*(2.*C2*XC+C3*XB)/(R*T)
c       ************************************
c       Once the LA model has converged to its
c       solution, activity coefficients are
c       obtained from the NRTL.
        IF(LCNVD.EQ.1)GAM(I)=DLNG(I)

        DGAM(I,I)=DG1
                DO 15 J=1,N
                        IF(I.EQ.J)GOTO15
                        DGAM(I,J)=DG3
                        IF(IPS(I,J).EQ.1)DGAM(I,J)=DG2
15              CONTINUE
20      CONTINUE
        RETURN
        END
```

```
      SUBROUTINE LAQUAT(X,GAM,DGAM,IPS,LCNVD)

c     A pseudo-quartenary two-suffix Margules
c     is used as the activity coefficient
c     correlation.
c     ****************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/OPT/NSS,KIND,NMT,NLA1
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/INIT/XLIM(10,10),IV,IL
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     & GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     & YPTS(50,10),DPTS(50,3)
      COMMON/UPDATE/INXTX,INXTY
      INTEGER IPTS(50),IPS(10,10)
      REAL*8 IB,IC,ID
      REAL*8 X(20),GAM(20),DGAM(20,20),DLNG(20)
      REAL*8 ILAGAM(20),SSGAM(20),SGAM(20,20)
c     ****************************************
      IFDER=0
      DATA R/1.987/
      IPHS=1
      IF(X(IV).GT.X(IL))IPHS=2
      IF(LCNVD.NE.1)GOTO4
      CALL NRTL(X,DLNG,DGAM)
c     ****************************************
c     After the LA model has converged to
c     its solution, activity coefficients
c     are then provided by the NRTL. These
c     additional phase compositions and
c     activity coefficients can be added
c     to the regression data. The
c     model can then be updated.

c     IF(IPHS.EQ.1.AND.INXTX.GT.66)GOTO4
c     IF(IPHS.EQ.2.AND.INXTY.GT.66)GOTO4
c     DO 1 I=1,N
c             IF(IPHS.EQ.1)XPTS(INXTX,I)=X(I)
c             IF(IPHS.EQ.2)YPTS(INXTY,I)=X(I)
c     IF(IPHS.EQ.1)GXPTS(INXTX,I)=R*T*DLOG(DLNG(I))
c     IF(IPHS.EQ.2)GYPTS(INXTY,I)=R*T*DLOG(DLNG(I))
c1    CONTINUE
c     IF(IPHS.EQ.1)NPTS=INXTX
c     IF(IPHS.EQ.2)NPTS=INXTY
c     IF(IPHS.EQ.1)INXTX=INXTX+1
c     IF(IPHS.EQ.2)INXTY=INXTY+1
c     DO 2 I=1,NPTS
c             IF(IPHS.EQ.1)IPTS(I)=I
```

```
c           IF(IPHS.EQ.2)IPTS(I)=I
c2      CONTINUE
4       DO 20 I=1,N
                XB=0.
                XC=0.
                XD=0.
                DO 5 J=1,N
                        IB=0.
                        IC=0.
                        ID=0.
                        IF(IPS(I,J).EQ.1)IB=1.
                        IF(IPS(I,J).EQ.2)IC=1.
                        IF(IPS(I,J).EQ.3)ID=1.
                        XB=XB+X(J)*IB
                        XC=XC+X(J)*IC
                        XD=XD+X(J)*ID
5               CONTINUE
        C1=PARMX(I,1)
        C2=PARMX(I,2)
        C3=PARMX(I,3)
        C4=PARMX(I,4)
        C5=PARMX(I,5)
        C6=PARMX(I,6)
        IF(IPHS.EQ.1)GOTO10
        C1=PARMY(I,1)
        C2=PARMY(I,2)
        C3=PARMY(I,3)
        C4=PARMY(I,4)
        C5=PARMY(I,5)
        C6=PARMY(I,6)
10      GAM(I)=DEXP(
     &           ((C1*XB**2)+(C2*XC**2)+(C3*XD**2)
     &         +(C4*XB*XC)+(C5*XB*XD)+(C6*XC*XD))
     &           /(R*T)
     &              )
        ILAGAM(I)=GAM(I)
c       ****************************************
c       The partial derivatives of the activity
c       coefficient with respect to xi,xb,xc,and
c       xd (DG1,DG2,DG3,DG4) are calculated.

        DG1=-2.*GAM(I)*DLOG(GAM(I))
        DG2=DG1+(GAM(I)/(R*T))*(2.*C1*XB+C4*XC+C5*XD)
        DG3=DG1+(GAM(I)/(R*T))*(2.*C2*XC+C4*XB+C6*XD)
        DG4=DG1+(GAM(I)/(R*T))*(2.*C3*XD+C5*XB+C6*XC)
c       ****************************************
c       Once the LA model has converged to its
c       solution, activity coefficients are
c       obtained from the NRTL.
```

```
      IF(LCNVD.EQ.1)GAM(I)=DLNG(I)

      DGAM(I,I)=DG1
               DO 15 J=1,N
                        IF(I.EQ.J)GOTO15
               IF(IPS(I,J).EQ.1)DGAM(I,J)=DG2
               IF(IPS(I,J).EQ.2)DGAM(I,J)=DG3
               IF(IPS(I,J).EQ.3)DGAM(I,J)=DG4
15             CONTINUE
20    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LAEXPTER(X,GAM,DGAM,IPS,LCNVD,BGAM)

c     The expanded pseudo-ternary 2-suffix Margules is
c     used to obtain activity coefficients.
c     ****************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/INIT/XLIM(10,10),IV,IL
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     & GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     & YPTS(50,10),DPTS(50,3)
      COMMON/OPT/NSS,KIND,NMT,NLA1
      COMMON/BASE/XBSTAR(20,20),XCSTAR(20,20),XDSTAR(20,20)
      COMMON/UPDATE/INXTX,INXTY
      INTEGER IPTS(50),IPS(10,10)
      REAL*8 BGAM(20,20)
      REAL*8 ILAGAM(20),YLAM(20,20),XLAM(20,20)
      REAL*8 II,IB,IC
      REAL*8 X(20),GAM(20),DGAM(20,20),DLNG(20)
      DATA R/1.987/
c     ****************************************
      IFDER=0
      IPHS=1
      IF(X(IV).GT.X(IL))IPHS=2

      DO 40 I=1,N
            XB=0.
            XC=0.
            DO 25 J=1,N
                  IB=0.
                  IC=0.
                  IF(IPS(I,J).EQ.1)IB=1.
                  IF(IPS(I,J).EQ.-1)IC=1.
                  XB=XB+X(J)*IB
                  XC=XC+X(J)*IC
25          CONTINUE
      C1=PARMX(I,1)
      C2=PARMX(I,2)
      C3=PARMX(I,3)
      C4=PARMX(I,4)
      C5=PARMX(I,5)
      IF(IPHS.EQ.1)GOTO30
      C1=PARMY(I,1)
      C2=PARMY(I,2)
      C3=PARMY(I,3)
      C4=PARMY(I,4)
      C5=PARMY(I,5)
c     ****************************************
c     Activity coefficient of cmpt of
```

```
c       interest is computed.

30      GAM(I)=   DEXP(
     &                  (
     &  1.987*T*DLOG(BGAM(I,IPHS))
     &  +C1*(XB-XBSTAR(I,IPHS))
     &  +C2*(XC-XCSTAR(I,IPHS))
     &  +C3*(XB-XBSTAR(I,IPHS))**2
     &  +C4*(XC-XCSTAR(I,IPHS))**2
     &  +C5*(XB-XBSTAR(I,IPHS))*(XC-XCSTAR(I,IPHS))
     &                      )/(1.987*T)
     &                                  )
c       ****************************************
c       Partial activity coefficient of cmpt
c       of interest with respect to its
c       mole number.

        DG1=(GAM(I)/(R*T))*(
     &  -C1*(XB)-C2*(XC)
     &  -2*C3*(XB)*(XB-XBSTAR(I,IPHS))
     &  -2*C4*(XC)*(XC-XCSTAR(I,IPHS))
     &  -C5*(XB)*(XC-XCSTAR(I,IPHS))
     &  -C5*(XC)*(XB-XBSTAR(I,IPHS))
     &                      )
c       ****************************************
c       Partial activity coeff. of cmpt of
c       interest w/r to pseudo-mole # B

        DG2= (GAM(I)/(R*T))*(
     &  C1*(1-XB)-C2*(XC)
     &  +2*C3*(1-XB)*(XB-XBSTAR(I,IPHS))
     &  -2*C4*(XC)*(XC-XCSTAR(I,IPHS))
     &  +C5*(1-XB)*(XC-XCSTAR(I,IPHS))
     &  -C5*(XC)*(XB-XBSTAR(I,IPHS))
     &                      )
c       ****************************************
c       Partial activity coeff. of cmpt of
c       interest w/r to pseudo-mole # C

        DG3= (GAM(I)/(R*T))*(
     &   -C1*(XB)+C2*(1-XC)
     &  -2*C3*(XB)*(XB-XBSTAR(I,IPHS))
     &  +2*C4*(1-XC)*(XC-XCSTAR(I,IPHS))
     &  -C5*(XB)*(XC-XCSTAR(I,IPHS))
     &  +C5*(1-XC)*(XB-XBSTAR(I,IPHS))
     &                      )
        DGAM(I,I)=DG1
                DO 35 J=1,N
                        IF(I.EQ.J)GOTO 35
```

```
                        DGAM(I,J)=DG3
                IF(IPS(I,J).EQ.1)DGAM(I,J)=DG2
35              CONTINUE
40      CONTINUE
        RETURN
        END
```

```
      SUBROUTINE LAEXPQUAT(X,GAM,DGAM,IPS,LCNVD,BGAM)

c     The expanded pseudo-quartenary 2-suffix
c     Margules is used to obtain activity
c     coefficients
c     ****************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/INIT/XLIM(10,10),IV,IL
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     & GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     & YPTS(50,10),DPTS(50,3)
      COMMON/OPT/NSS,KIND,NMT,NLA1
      COMMON/BASE/XBSTAR(20,20),XCSTAR(20,20),XDSTAR(20,20)
      COMMON/UPDATE/INXTX,INXTY
      INTEGER IPTS(50),IPS(10,10)
      REAL*8 BGAM(20,20)
      REAL*8 ILAGAM(10),SGAM(20,20),YLAM(20,20),XLAM(20,20)
      REAL*8 IB,IC,ID
      REAL*8 X(20),GAM(20),DGAM(20,20),DLNG(20)
      DATA R/1.987/
c     ****************************************
      IFDER=0
      IPHS=1
      IF(X(IV).GT.X(IL))IPHS=2

      DO 40 I=1,N
            XB=0.
            XC=0.
            XD=0.
            DO 25 J=1,N
                  IB=0.
                  IC=0.
                  ID=0.
                  IF(IPS(I,J).EQ.1)IB=1.
                  IF(IPS(I,J).EQ.2)IC=1.
                  IF(IPS(I,J).EQ.3)ID=1.
                  XB=XB+X(J)*IB
                  XC=XC+X(J)*IC
                  XD=XD+X(J)*ID
25          CONTINUE
      C1=PARMX(I,1)
      C2=PARMX(I,2)
      C3=PARMX(I,3)
      C4=PARMX(I,4)
      C5=PARMX(I,5)
      C6=PARMX(I,6)
      C7=PARMX(I,7)
      C8=PARMX(I,8)
```

```
      C9=PARMX(I,9)
      IF(IPHS.EQ.1)GOTO 30
      C1=PARMY(I,1)
      C2=PARMY(I,2)
      C3=PARMY(I,3)
      C4=PARMY(I,4)
      C5=PARMY(I,5)
      C6=PARMY(I,6)
      C7=PARMY(I,7)
      C8=PARMY(I,8)
      C9=PARMY(I,9)
c     ****************************************
c     Activity coefficient of cmpt of
c     interest is computed.

30    GAM(I)=   DEXP(
     &                 (
     &  1.987*T*DLOG(BGAM(I,IPHS))
     &  +C1*(XB-XBSTAR(I,IPHS))
     &  +C2*(XC-XCSTAR(I,IPHS))+C3*(XD-XDSTAR(I,IPHS))
     &  +C4*(XB-XBSTAR(I,IPHS))**2
     &  +C5*(XC-XCSTAR(I,IPHS))**2+C6*(XD-XDSTAR(I,IPHS))**2
     &  +C7*(XB-XBSTAR(I,IPHS))*(XC-XCSTAR(I,IPHS))
     &  +C8*(XB-XBSTAR(I,IPHS))*(XD-XDSTAR(I,IPHS))
     &  +C9*(XC-XCSTAR(I,IPHS))*(XD-XDSTAR(I,IPHS))
     &                      )/(1.987*T)
     &                                )
c     ****************************************
c     Partial activity coefficient of cmpt
c     of interest with respect to its
c     mole number.

      DG1= (GAM(I)/(R*T))*
     &       (
     &   -C1*(XB)-C2*(XC)-C3*(XD)
     &  -2*C4*(XB)*(XB-XBSTAR(I,IPHS))
     &  -2*C5*(XC)*(XC-XCSTAR(I,IPHS))
     & -2*C6*(XD)*(XD-XDSTAR(I,IPHS))
     &  -C7*(XB)*(XC-XCSTAR(I,IPHS))
     &-C7*(XC)*(XB-XBSTAR(I,IPHS))
     &  -C8*(XB)*(XD-XDSTAR(I,IPHS))
     &-C8*(XD)*(XB-XBSTAR(I,IPHS))
     &  -C9*(XC)*(XD-XDSTAR(I,IPHS))
     &-C9*(XD)*(XC-XCSTAR(I,IPHS))
     &       )
c     ****************************************
c     Partial activity coeff. of cmpt of
c     interest w/r to pseudo-mole # B
```

```
      DG2=(GAM(I)/(R*T))*
     &          (
     &   C1*(1-XB)-C2*(XC)-C3*(XD)
     &  +2*C4*(1-XB)*(XB-XBSTAR(I,IPHS))
     &  -2*C5*(XC)*(XC-XCSTAR(I,IPHS))
     &  -2*C6*(XD)*(XD-XDSTAR(I,IPHS))
     &  +C7*(1-XB)*(XC-XCSTAR(I,IPHS))
     &  -C7*(XC)*(XB-XBSTAR(I,IPHS))
     &  +C8*(1-XB)*(XD-XDSTAR(I,IPHS))
     &  -C8*(XD)*(XB-XBSTAR(I,IPHS))
     &  -C9*(XC)*(XD-XDSTAR(I,IPHS))
     &  -C9*(XD)*(XC-XCSTAR(I,IPHS))
     &          )
C       ****************************************
C       Partial activity coeff. of cmpt of
C       interest w/r to pseudo-mole # C

      DG3=(GAM(I)/(R*T))*
     &          (
     &  -C1*(XB)+C2*(1-XC)-C3*(XD)
     &  -2*C4*(XB)*(XB-XBSTAR(I,IPHS))
     &  +2*C5*(1-XC)*(XC-XCSTAR(I,IPHS))
     &  -2*C6*(XD)*(XD-XDSTAR(I,IPHS))
     &  -C7*(XB)*(XC-XCSTAR(I,IPHS))
     &  +C7*(1-XC)*(XB-XBSTAR(I,IPHS))
     &  -C8*(XB)*(XD-XDSTAR(I,IPHS))
     &  -C8*(XD)*(XB-XBSTAR(I,IPHS))
     &  -C9*(XD)*(XC-XCSTAR(I,IPHS))
     &  +C9*(1-XC)*(XD-XDSTAR(I,IPHS))
     &          )
C       ****************************************
C       Partial activity coeff. of cmpt of
C       interest w/r to pseudo-mole # D

      DG4=(GAM(I)/(R*T))*
     &          (
     &  -C1*(XB)-C2*(XC)+C3*(1-XD)
     &  -2*C4*(XB)*(XB-XBSTAR(I,IPHS))
     &  -2*C5*(XC)*(XC-XCSTAR(I,IPHS))
     &  +2*C6*(1-XD)*(XD-XDSTAR(I,IPHS))
     &  -C7*(XB)*(XC-XCSTAR(I,IPHS))
     &  -C7*(XC)*(XB-XBSTAR(I,IPHS))
     &  -C8*(XB)*(XD-XDSTAR(I,IPHS))
     &  +C8*(1-XD)*(XB-XBSTAR(I,IPHS))
     &  -C9*(XC)*(XD-XDSTAR(I,IPHS))
     &  +C9*(1-XD)*(XC-XCSTAR(I,IPHS))
     &          )
C       ****************************************
C       The n x n derivative matrix of LA model
```

```
c       is set up.

        DGAM(I,I)=DG1
                 DO 35 J=1,N
                 IF(I.EQ.J)GOTO35
                 IF(IPS(I,J).EQ.1)DGAM(I,J)=DG2
                 IF(IPS(I,J).EQ.2)DGAM(I,J)=DG3
                 IF(IPS(I,J).EQ.3)DGAM(I,J)=DG4
35               CONTINUE
40      CONTINUE
        RETURN
        END
```

```
      SUBROUTINE LAMODEL(X,GAM,DGAM,IPS,LCNVD,BGAM)

c     A linearized Taylor series expansion
c     of the 2-suffix Margules equation is
c     used a an activity coefficient correlation.
c     ****************************************
      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/BASE1/X1S(10,10),X2S(10,10),X3S(10,10),X4S(10,10),
     &  X5S(10,10),X6S(10,10)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/INIT/XLIM(10,10),IV,IL
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     &  GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     &  YPTS(50,10),DPTS(50,3)
      COMMON/OPT/NSS,KIND,NMT,NLA1
      COMMON/UPDATE/INXTX,INXTY
      INTEGER IPS(10,10)
      REAL*8 BGAM(20,20)
      REAL*8 X(20),GAM(20),DGAM(20,20),DLNG(20)
      DATA R/1.987/
c     ****************************************
      IPHS=1
      IF(X(IV).GT.X(IL))IPHS=2

      X1=X(1)
      X2=X(2)
      X3=X(3)
      X4=X(4)
      X5=X(5)
      X6=X(6)
      DO 40 I=1,N
      C1=PARMX(I,1)
      C2=PARMX(I,2)
      C3=PARMX(I,3)
      C4=PARMX(I,4)
      C5=PARMX(I,5)
      C6=PARMX(I,6)
      IF(IPHS.EQ.1)GOTO30
      C1=PARMY(I,1)
      C2=PARMY(I,2)
      C3=PARMY(I,3)
      C4=PARMY(I,4)
      C5=PARMY(I,5)
      C6=PARMY(I,6)
c     ****************************************
c     Activity coefficient of cmpt of
c     interest is computed.

30    GAM(I)=   DEXP(
```

```
     &                    (
     &  1.987*T*DLOG(BGAM(I,IPHS))
     &  +C1*(X1-X1S(I,IPHS))
     &  +C2*(X2-X2S(I,IPHS))
     &  +C3*(X3-X3S(I,IPHS))+C4*(X4-X4S(I,IPHS))
     &  +C5*(X5-X5S(I,IPHS))
     &  +C6*(X6-X6S(I,IPHS))
     &                      )/(1.987*T)
     &                   )
c       ****************************************
c       Partial activity coefficient of cmpt
c       of interest with respect to its
c       mole number.

        DG1= (-GAM(I)/(R*T))*
     &         (
     &   C1*(X1-1.)+
     &   C2*X2+C3*X3+C4*X4+C5*X5+C6*X6
     &         )
c       ****************************************
c       Partial activity coeff. of cmpt of
c       interest w/r to mole # 2

        DG2=(-GAM(I)/(R*T))*
     &          (
     &   C1*X1+
     &   C2*(X2-1.)+C3*X3+C4*X4+C5*X5+C6*X6
     &          )
c       ****************************************
c       Partial activity coeff. of cmpt of
c       interest w/r to mole # 3

        DG3=(-GAM(I)/(R*T))*
     &          (
     &   C1*X1+
     &   C2*X2+C3*(X3-1.)+C4*X4+C5*X5+C6*X6
     &          )
c       ****************************************
c       Partial activity coeff. of cmpt of
c       interest w/r to mole # 4

        IF(N.LT.4)GOTO 38
        DG4=(-GAM(I)/(R*T))*
     &          (
     &   C1*X1+
     &   C2*X2+C3*X3+C4*(X4-1.)+C5*X5+C6*X6
     &          )
c       ****************************************
c       Partial activity coeff. of cmpt of
```

```
c       interest w/r to mole # 5

        IF(N.LT.5)GOTO 38
        DG5=(-GAM(I)/(R*T))*
     &          (
     &   C1*X1+
     &   C2*X2+C3*X3+C4*X4+C5*(X5-1.)+C6*X6
     &          )
c       ****************************************
c       Partial activity coeff. of cmpt of
c       interest w/r to mole # 6

        IF(N.LT.6)GOTO 38
        DG6=(-GAM(I)/(R*T))*
     &          (
     &   C1*X1+
     &   C2*X2+C3*X3+C4*X4+C5*X5+C6*(X6-1.)
     &          )
c       ****************************************
c       The n x n derivative matrix of LA model
c       is set up.

38      DGAM(I,1)=DG1
        DGAM(I,2)=DG2
        DGAM(I,3)=DG3
        DGAM(I,4)=DG4
        DGAM(I,5)=DG5
        DGAM(I,6)=DG6
40      CONTINUE
        RETURN
        END
```

```
      SUBROUTINE DIST(NPTS)

c     The distance of a composition point
c     from the base point is taken. The base
c     point is either the most recent value
c     of the x and y in the iteration or the
c     point around which regression
c     compositions have been obtained.
c     ****************************************

      IMPLICIT REAL*8 (A-H,O-Z)
      COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
     & DGAMX(20,20),DGAMY(20,20)
      COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
      COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
     & GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
     & YPTS(50,10),DPTS(50,3)
c     ****************************************
      WRITE(6,*)
      WRITE(6,*)'XPTS DISTANCES'
      DO 10 J=1,NPTS
      D=0.
      DD=0.
      DDD=0.
      DO 5 K=1,N
            D=D+(X(K)-XPTS(J,K))**2
5     CONTINUE
      DD=DSQRT(D)
      DDD=100.*DD
      WRITE(6,*)'DISTANCE OF POINT ',J
     & ,' FROM B.P. IS ',  DDD,' %'
10    CONTINUE
      WRITE(6,*)'YPTS DISTANCES'
      DO 20 J=1,NPTS
      D=0.
      DD=0.
      DDD=0.0
      DO 15 K=1,N
            D=D+(Y(K)-YPTS(J,K))**2
15    CONTINUE
      DD=DSQRT(D)
      DDD=100.*DD
      WRITE(6,*)'DISTANCE OF POINT ',J
     & ,' FROM B.P. IS ',  DDD,' %'
20    CONTINUE
      RETURN
      END
```

```
c       SUBROUTINE UPDAT(IPHS)

c       ****************************************
c       IMPLICIT REAL*8 (A-H,O-Z)
c       COMMON/CONV/X(20),Y(20),A,RK(20),GAMX(20),GAMY(20),
c&      DGAMX(20,20),DGAMY(20,20)
c       P IS REMOVED FROM MT-THE LAST TERM
c       COMMON/MT/C(20,20),ALPHA(20,20),N,T,IFDER,Z(20),P
c       COMMON/INIT/XLIM(10,10),IV,IL
c       COMMON/LOCAL/IPSX(10,10),IPSY(10,10),GXPTS(50,10),
c&      GYPTS(50,10),PARMX(10,15),PARMY(10,15),XPTS(50,10),
c&      YPTS(50,10),DPTS(50,3)
c       COMMON/UPDATE/INXTX,INXTY
c       DIMENSION IPTS(50)
c       DIMENSION TPT(10)
c       ****************************************
c       IF(IPHS.EQ.2)GOTO45
c       ****************************************
c       CONSIDER X PHASE
c       ****************************************
c       CALL POINTS(X)
c       S=1.-(X(IV)+X(IL))
c       DO 40 I=1,5
c               TPT(IV)=DPTS(I,2)
c               TPT(IL)=DPTS(I,3)
c               DO 10 J=1,N
c                       IF(J.EQ.IV.OR.J.EQ.IL)GOTO10
c                       TPT(J)=DPTS(I,1)*X(J)/S
c10             CONTINUE
c               DIST=.02
c               DD=.025
c               IRP=0
c               DO 20 J=1,50
c                       IF(J.GE.INXTX)GOTO25
c                       D=0.
c                       DO 15 K=1,N
c                               D=D+(XPTS(J,K)-TPT(K))**2
c15                     CONTINUE
c                       D=DSQRT(D)
c                       IF(D.GT.DIST)GOTO 20
c                       IF(D.GT.DD)GOTO20
c                       DIST=D
c                       DD=D
c                       IRP=J
c20             CONTINUE
c25             IF(IRP.EQ.0)GOTO30
c               IPTS(I)=IRP
c               GOTO40
c30             DO 35 J=1,N
```

```
c                       XPTS(INXTX,J)=TPT(J)
c35               CONTINUE
c                 IPTS(I)=INXTX
c                 INXTX=INXTX+1
c40       CONTINUE
c         CALL NEWMOD(IPTS,IPHS,5)
c         RETURN
c         ****************************************
c         Y PHASE IS NOW CONSIDERED
c         ****************************************
c45       CALL POINTS(Y)
c         S=1.-(Y(IV)+Y(IL))
c         DO 80 I=1,5
c                 TPT(IV)=DPTS(I,2)
c                 TPT(IL)=DPTS(I,3)
c                 DO 50 J=1,N
c                         IF(J.EQ.IV.OR.J.EQ.IL)GOTO50
c                         TPT(J)=DPTS(I,1)*Y(J)/S
c50               CONTINUE
c                 DIST=.02
c                 DD=.025
c                 IRP=0
c                 DO 60 J=1,50
c                         IF(J.GE.INXTY)GOTO65
c                         D=0.
c                         DO 55 K=1,N
c                                 D=D+(YPTS(J,K)-TPT(K))**2
c55                       CONTINUE
c                         D=DSQRT(D)
c                         IF(D.GT.DIST)GOTO 60
c                         IF(D.GT.DD)GOTO60
c                         DIST=D
c                         DD=D
c                         IRP=J
c60               CONTINUE
c65               IF(IRP.EQ.0)GOTO70
c                 IPTS(I)=IRP
c                 GOTO80
c70               DO 75 J=1,N
c                         YPTS(INXTY,J)=TPT(J)
c75               CONTINUE
c                 IPTS(I)=INXTY
c                 INXTY=INXTY+1
c80       CONTINUE
c         CALL NEWMOD(IPTS,IPHS,5)
c         RETURN
c         END
```