ABSTRACT

Title of Thesis: Programming for the Prediction of Thermodynamic Phenomena.

One Kwon Rim, Master of Science in Chemical Engineering, 1988

Thesis directed by Dr. Dana E. Knox

A Thermodynamic Calculation Program was developed. The program contains the calculation procedure for VAPOR/LIQUID EQUILIBRIUM(Isothermal P-X data and T-X data), HEAT-OF-MIXING(liquid mixture data) and LIQUID/LIQUID EQUILIBRIUM SEPARATION data. The program may optionally use the Knox Molecular(GAM), UNIQUAC, NRTL, Wilson and Redlich-Kister equations, and is prepared to accept any kind of model which will be developed in the future. This program also performs parameter file creation, parameter input, updating of parameters, and is structured to allow implementation of data regression when program is updated. Comparison and analysis of each function were executed using this program. It was revealed that VLE(Vapor-Liquid Equilibrium) predictions are most easily made, while LLES (Liquid-Liquid Equilibrium Separation) predictions can be performed in limited ranges and Heat-of-mixing(Liquid mixture) predictions showed relatively poor results. The models which were used to predict the above were also compared and analyzed with each other. The Knox molecular(GAM) and

modified UNIQUAC equations showed the most best performance, but each model has its advantages and disadvantages. It was concluded that all systems can not be treated by only one model. Each system must select the most suitable model which fits well for its system. Eventually the model which can perform all predictions using the same parameter values must be developed. Also, there is a need for a model which can also treat electrolyte systems.

PROGRAMMING FOR THE PREDICTION

OF

THERMODYNAMIC PHENOMENA

BY
ONE KWON RIM

Thesis submitted to the faculty of the Graduate School of
the New Jersey Institute of Technology in partial
fulfillment of the requirements for the degree of
Master of Science in Chemical Engineering
1988

APPROVAL SHEET

Title of Thesis: Programming for the Prediction
of Thermodynamic Phenomena.

Name of Candidate: One Kwon Rim
Master of Science in Chemical
Engineering, 1988

Thesis and Abstract Approved:

Dana E. Knox Date
Assistant Professor
Dept. of Chemical
Engineering, Chemistry,
and Environmental Science

Edward C. Roche Date
Professor
Dept. of Chemical
Engineering, Chemistry,
and Environmental Science

John E. McCormick Date
Professor
Dept. of Chemical
Engineering, Chemistry,
and Environmental Science

VITA

Name: One Kwon Rim.

Permanent Address:

Degree and date to be conferred: Master of Science in Chemical Engineering, 1988.

Date of Birth:

Place of Birth:

Collegiate institutions attended	Dates	Degree	Date
Seoul National University, Korea.	75-79	B.S	Aug., 79.
Seoul National University, Korea.	80-82	M.S	Feb., 82.
New Jersey Institute of Technology.	85-88	M.S.Ch.E	May., 88.

Major: Chemical Engineering.

Acknowledgement

The author wishes to acknowledge the helpful discussions and encouragement offered by author's advisor, Dr. Dana E. Knox., and also wishes to acknowledge the helpful corrections by Dr. Edward C. Roche and Dr. John E. McCormick. In addition supports from his wife, Hey-Gyeong, and his relatives are gratefully acknowledged.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

I. Introduction

Separation of fluid mixtures is one of the main tasks of chemical engineering. These separations are usually performed by phase contacting operations: distillation, absorption, stripping and extraction. Design of equipment for such operations requires quantitative estimates of phase equilibria.

This thesis presents a detailed discussion of a method for computer-implemented estimation of phase equilibria in multicomponent systems. Attention is restricted to nonelectrolytes, i.e. organic liquids(hydrocarbons and their derivatives, alcohols, nitriles, ketones, esters, etc.) and a few common inorganic fluids such as water etc. Attention is also limited to low or moderate pressure and to conditions remote from critical.

A. Need for a Thermodynamic Treatment

The possible number of liquid and vapor mixtures in technological processes is incredibly large, and it is unreasonable to expect that experimental thermodynamic data will ever be available for a significant fraction of this number. Further, obtaining good experimental data requires appreciable experimental skill, experience, and patience. It is, therefore, an economic necessity to consider techniques for predicting data for multicomponent mixtures from few experimental data. Such a technique should require only a limited experimental effort and, whenever possible, should be based on a theoretical foundation to provide a reliability for interpolation and extrapolation with respect to temperature, pressure, and composition.

Vapor/liquid equilibria, liquid/liquid equilibria and heats of mixing in liquid mixtures each depend on the nature of the components present, on their concentrations in both phases, and on the temperature and pressure of the system. Because of the large number of variables which determine multicomponent equilibria, it is essential to use an efficient organizational tool which reduces available experimental data to a small number of theoretically significant functions and parameters; these functions and parameters may then be called on to form the building blocks on which to construct the desired equilibria. Such an organizational tool is provided by thermodynamic analysis and synthesis. First, limited pure-component and binary data are analyzed to yield fundamental thermodynamic quantities. Second, these quantities are reduced to obtain parameters in a molecular model. In this thesis, these steps are skipped, and so parameters must be supplied by external sources references. That model, by synthesis, may be used to calculate the phase behavior of multicomponent liquids and vapors. In this way, it is possible to "scale up" available data on binary and pure-component systems in order to obtain good estimates of the properties of multicomponent mixtures of a large variety of components including water, polar organic solvents such as ketones, alcohols, nitriles, etc., and paraffinic, naphthenic, and aromatic hydrocarbons. In this thesis, model behaviors are programmed so that complicated hand calculation procedures can be eliminated. This is the main utility of the thesis.

B. Theoretical Background

The method proposed in this thesis has a firm thermodynamic basis. For vapor-liquid equilibria, the method may be used at low or moderate pressures commonly encountered in separation operations since vapor phase nonidealities are taken into account. For liquid-liquid equilibria and for heat of mixing of liquid mixtures, the effect of pressure is usually not important unless the pressure is very large or unless conditions are near the vapor-liquid critical region.

The detailed techniques presented here are based on models for the vapor phase (Hayden-O'Connell) and for the liquid phase(Knox, UNIQUAC, modified UNIQUAC, NRTL, Wilson and Redlich-Kister).

In vapor-liquid equilibria, if the composition of one phase is given, there are basically four types of problems, characterized by those variables which are specified and those which are to be calculated. Let T stand for temperature, P for total pressure, x_i for the mole fraction of component i in the liquid phase, and y_i for the mole fraction of component i in the vapor phase. For a mixture containing m components, the four types can be organized in this way :

Given	Find
$P, x_1, x_2, \ldots\ldots x_m$	$T, y_1, y_2, \ldots\ldots y_m$
$T, x_1, x_2, \ldots\ldots x_m$	$P, y_1, y_2, \ldots\ldots y_m$
$P, y_1, y_2, \ldots\ldots y_m$	$T, x_1, x_2, \ldots\ldots x_m$
$T, y_1, y_2, \ldots\ldots y_m$	$P, x_1, x_2, \ldots\ldots x_m$

In each of these problems, there are m unknowns; either the pressure or the temperature is unknown and there are m-1 unknown mole fractions. In this thesis, the 3rd and 4th cases are treated. When only the total system composition, pressure, and temperature (or enthalpy) are specified, the problem becomes a flash calculation.

In liquid-liquid equilibria, the total compositions and temperature are known; the pressure is usually not important. This problem is similar in some ways to a vapor-liquid flash and here is referred to as a liquid-liquid flash calculation.

The heat of mixing or enthalpy of mixing for a liquid solution also can not be calculated accurately without specific data for that solution. Once again, such data are correlated by means of a model such as Wilson, UNIQUAC, ..., etc.

The following are detailed procedures for each calculation.

1. Vapor-Liquid equilibrium calculation

For a vapor phase (superscript V) and a liquid phase (superscript L), at the same temperature, the equation of equilibrium for each component i is expressed in terms of the fugacity f_i:

$$f_i^V=f_i^L \qquad \text{(I-B-1.1)}$$

Equation (1) must be converted by the expression of accessible quantities and this can be done by introducing two auxiliary functions. The first one is the fugacity coefficient Φ, which relates the vapor-phase fugacity f_i^V to the mole fraction y_i and to the total pressure P. It is defined by

$$\Phi_i=f_i^V/(P*y_i) \qquad \text{(I-B-1.2)}$$

The activity coefficient γ relates the liquid-phase fugacity f_i^L to the mole fraction x_i and to a standard-state fugacity f_i^{0L}. The activity coefficent is defined by

$$\gamma_i=f_i^L/(x_i*f_i^{0L}) \qquad \text{(I-B-1.3)}$$

From Eqs. (1), (2), and (3) the equation of equilibrium for component i becomes

$$\Phi_i*y_i*P=\gamma_i*x_i*f_i^{0L} \qquad \text{(I-B-1.4)}$$

Equation (4) is the key equation for calculation of multicomponent vapor-liquid equilibria.

2. Liquid-liquid equilibria

For a liquid phase (superscript α) in equilibrium with another liquid phase (superscript β), the equation analogous to Equation (1) is

$$f_i^{\alpha}=f_i^{\beta} \qquad \text{(I-B-2.1)}$$

When the same standard-state fugacity is used in both phases, Equation (1) becomes

$$(\gamma_i*x_i)^{\alpha}=(\gamma_i*x_i)^{\beta} \qquad \text{(I-B-2.2)}$$

Equation (2) is the key equation for calculation of multicomponent liquid-liquid equilibria.

3. Heat of mixing

The general form of the Gibbs-Helmholtz equation is

$$\left[\frac{\partial\ g/T}{\partial\ 1/T} \right]_{P,x} = h \qquad \text{(I-B-3.1)}$$

where P, x indicates that both pressure and composition are held constant. If we apply this equation to a liquid mixture, it is convenient to rewrite the equation in the form

$$\left[\frac{\partial\ g^E/T}{\partial\ 1/T} \right]_{P,x} = h^E \qquad \text{(I-B-3.2)}$$

where g^E is the molar excess Gibbs energy of the mixture and h^E is the molar excess enthalpy, also called the enthalpy of mixing. Although it is always possible to differentiate an expression for the excess Gibbs energy with respect to temperature to find the enthalpy of mixing, such a procedure is often not reliable because the temperature dependence of the binary -mixture parameter of the liquid phase model is usually not known. As a result, heat of mixing data is generally regressed for a set of h^E parameters, independent of any g^E data or parameters.

II. Description of Program

This program is written in FORTRAN 77 and consists of one main program and 25 subroutines, giving a current total of approximately 3500 lines for this program. It performs both prediction of properties and parameter control, and is structured for the execution of data regression when the program is updated.

The prediction of properties is classified into four types of calculations. The first is Vapor/Liquid equilibrium (Isothermal P-X) data calculation. The second is Vapor/Liquid equilibrium(Isothermal T-X) data calculation. The third is Heat of Mixing calculation and the fourth is Liquid/Liquid equilibrium separation calculation. The parameter control function is divided into two categories. One is general parameter control which supplies pure component properties and the other is the binary interaction parameter control program. The following are individual descriptions of the various routines.

A. Main Program

The main program, MASTER, controls all selections which are made by the user. The user generally encounters four kinds of selection menus:

1. Goal Selection Menu.
2. Property Selection Menu.
3. Model Selection Menu.

4. Mixture and Components Selection Menu.

Each of these selections procedures which appear on the screen are represented as figures on the subsequent pages.

At present, choices 2),3) and 4) are available in Figure II-A-1. If 2) is chosen, the "Property Selection Menu" is waiting. If 3) is chosen it calls the subroutine, SORT, and returns to the "Goal Selection Menu" afterwards. If option 4) is chosen, the program is terminated.

Figure II-A-2 is the "Property Selection Menu". If the user chooses an option between 1) and 4), then the "Model Selection Menu" appears. Option 5) returns to the "Goal Selection Menu".

Figure II-A-3 is the "Model Selection Menu". Among these 12 options in Figure II-A-3, only 1), 5), 6), 8), 9), 0) and B) are currently available. If B) is chosen the user returns to the "Property Selection Menu". If one of the other available option is selected, the user arrives at the "Mixture and Component Selection Menu". This is similar to the previous cases. In this menu, there are 468 components which may be selected for the calculation. Of these, 25 components are currently available.

After the user has completed the menu selections, the subroutine SORT, which oversees the execution of all the previously determined options, is called. At that point, the program is either stopped or control is returned to the

"Goal Selection Menu". Fig.II-A-4 is the flow chart of program MASTER.

```
GOAL SELECTION MENU

Do you wish to:

1) Regress data
2) Predict properties
3) Control parameters
4) Quit program


Enter choice -->
```

Figure II-A-1. Goal Selection Menu.

```
PROPERTY SELECTION MENU

Are you interested in :

1) Vapor/Liquid Equilibrium(Isothermal P-X) data.
2) Vapor/Liquid Equilibrium(Isothermal T-X) data.
3) Heat-of-Mixing(Liquid mixture) data.
4) Liquid/Liquid Equilibrium Separation data.
5) Return to Goal Selection Menu.


Enter choice -->
```

Figure II-A-2. Property Selection Menu.

```
Model Selection Menu #1

Which model would you like :

1) Knox Molecular (GAM) Model.
2) Knox Group (GAG) Model.
3) Knox Association Model.
4) Knox Association Group Model.
5) UNIQUAC Model.
6) Modified UNIQUAC Model.
7) UNIFAC Model.
8) NRTL Model.
9) Wilson Model.
0) Redlich-Kister Model.
A) Access Menu #2
B) Return to Property Selection Menu.
```

Figure II-A-3. Model Selection Menu.

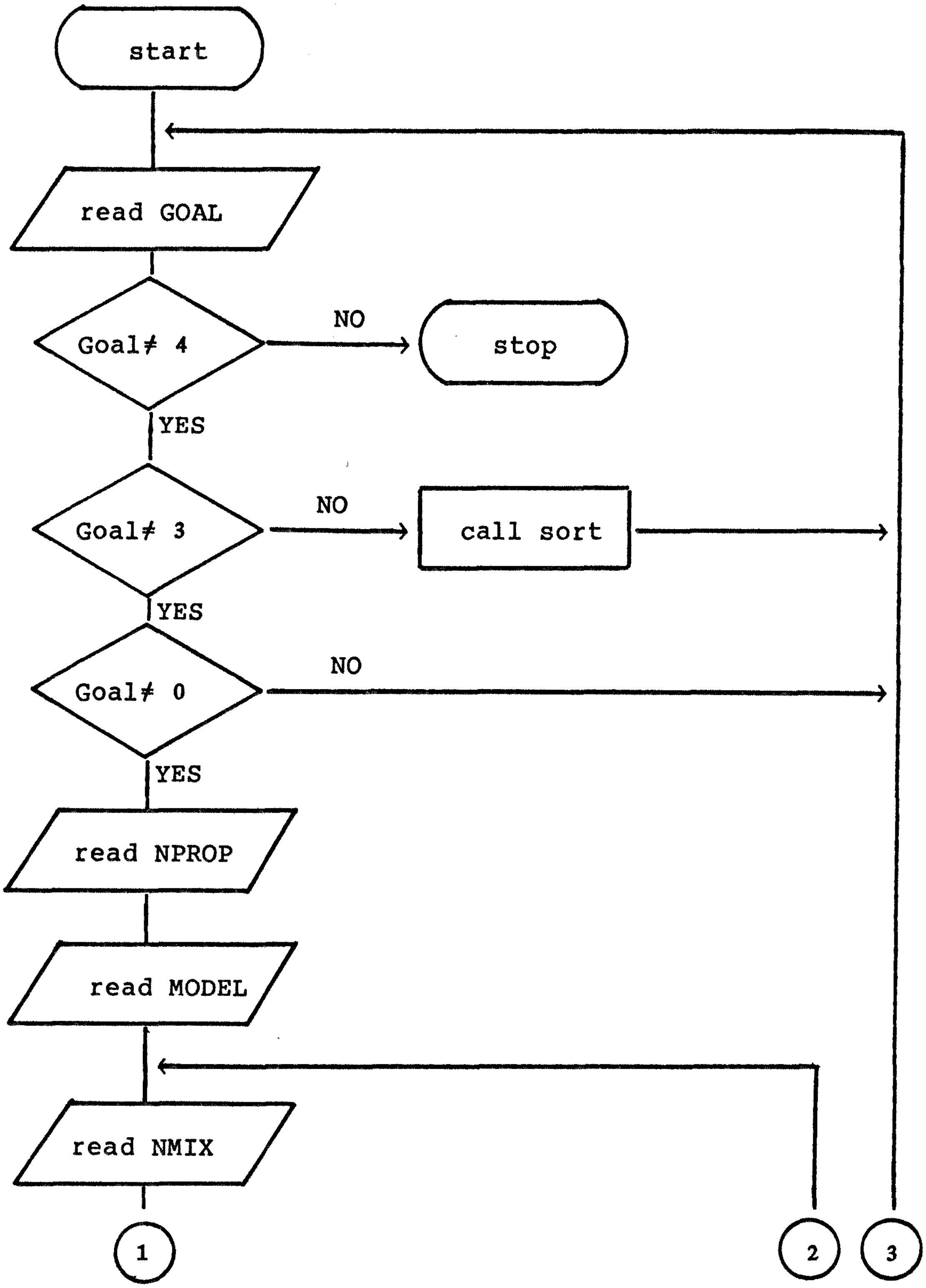

Figure II-A-4-a. Flow chart of the program MASTER.

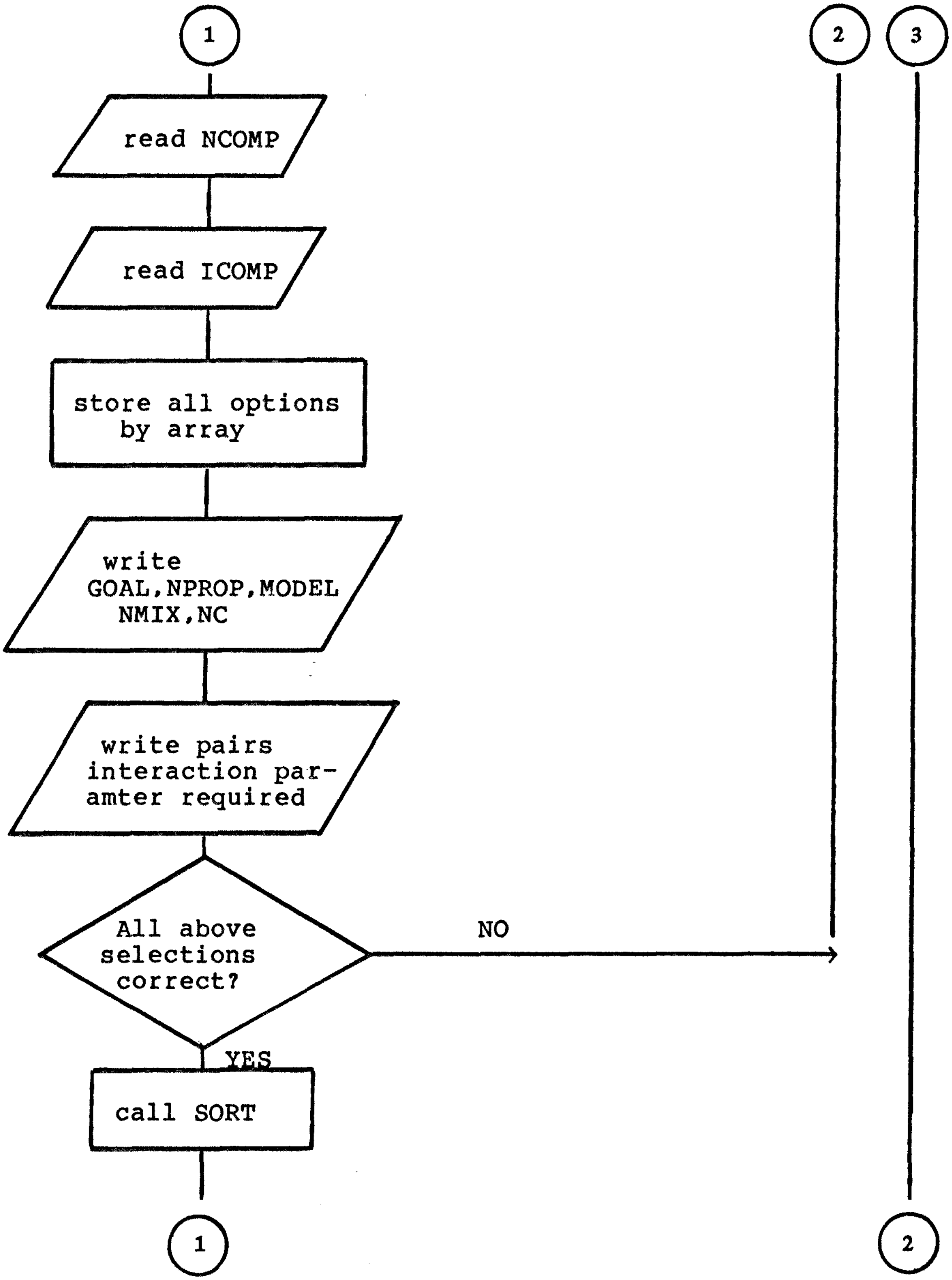
1
2
3
read NCOMP
read ICOMP
store all options by array
write GOAL,NPROP,MODEL NMIX,NC
write pairs interaction par-amter required
All above selections correct?
NO
YES
call SORT
1
2

Figure II-A-4-b. Flow chart of the program MASTER.

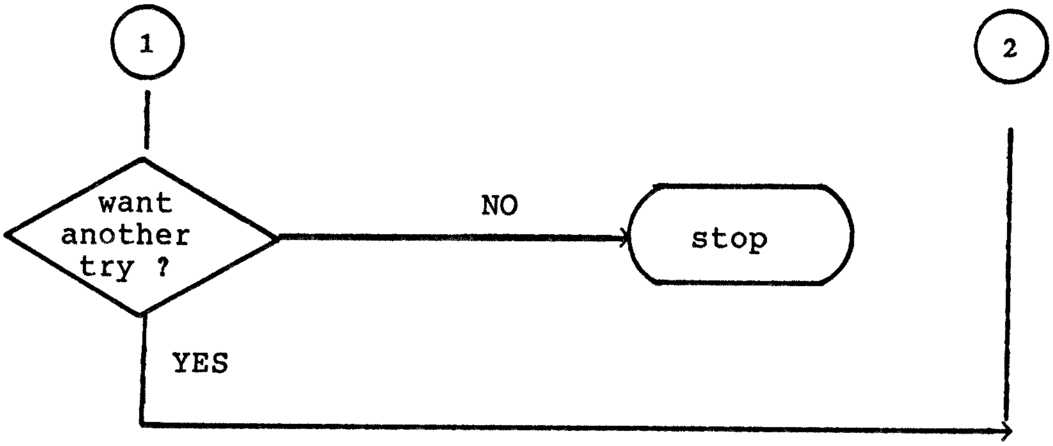

Figure II-A-4-c. Flow chart of the program MASTER.

B. Subroutine Programs

1. SORT.

Subroutine SORT is the program which examines the options selected in the main program and calls other subroutines to execute the chosen options. Subroutine SORT includes five subroutines. Their names are DEWP, DEWT, ELIPS, MIXHE and PARA. All of these subroutines function independently but they do use common subprograms. Each of these first four subroutines corresponds to a particular option in the "Property Selection Menu":

- DEWP to choice "1) Vapor/Liquid Equilibrium (P-x) Data"
- DEWT to choice "2) Vapor/Liquid Equilibrium (T-x) Data"
- MIXHE to choice "3) Heat-of-Mixing Data"
- ELIPS to choice "4) Liquid/Liquid Equilibrium Data"

Each of these four choices in turn corresponds to the option "2) Predict Properties" in the " Goal Selection Menu".

Subroutine PARA corresponds to the selection of "3) Parameter Control" in the "Goal Selection Menu".

This program is described by the flow chart in figure II-B-1.

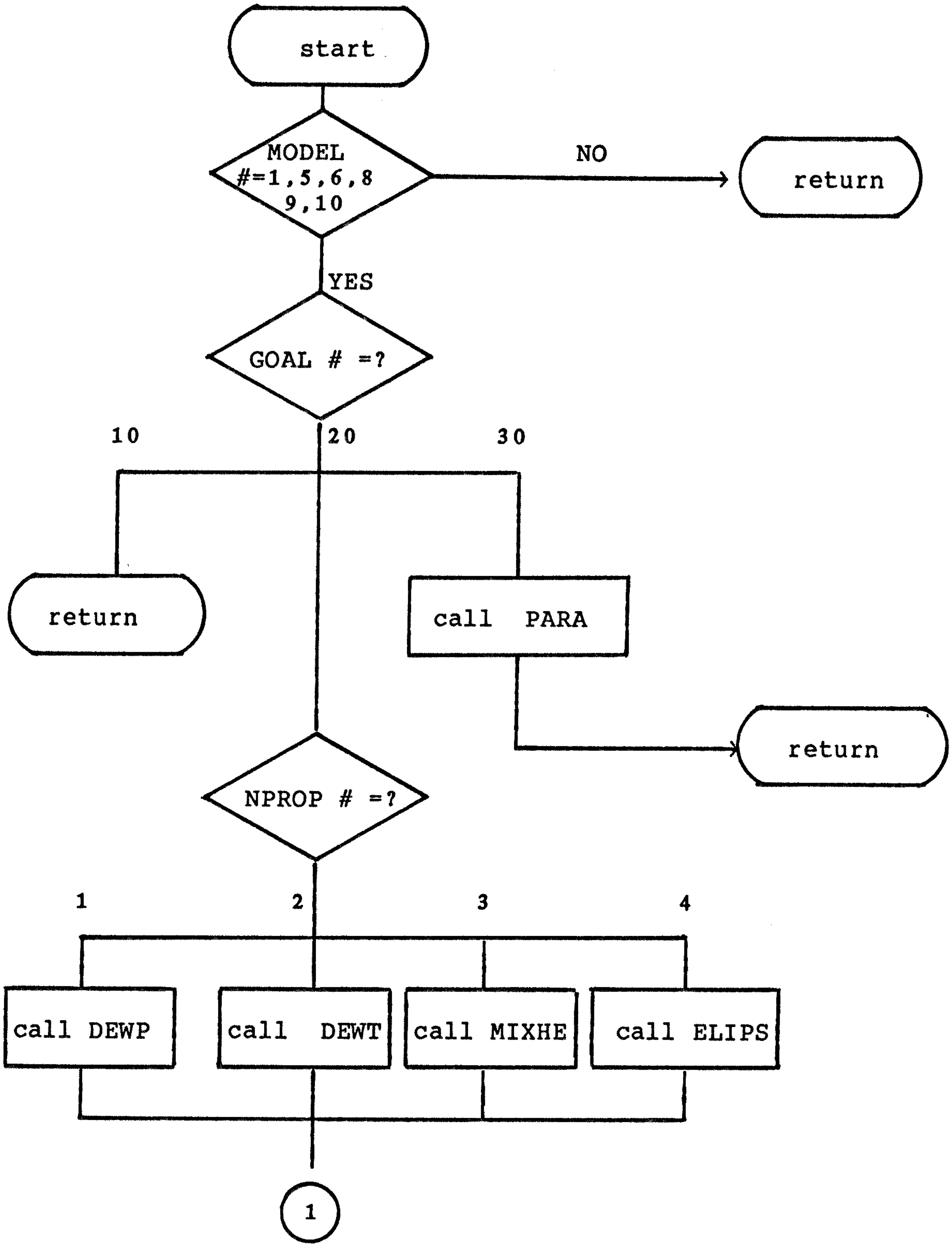

Figure II-B-1-a. Flow chart of the subroutine SORT.

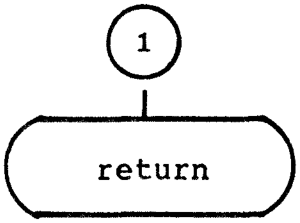

Figure II-B-1-b. Flow chart of the subroutine SORT.

2. DEWP.

Subroutine DEWP is the subprogram that calculates the dewpoint pressure. A dewpoint pressure calculation is a Vapor/Liquid Equilibrium calculation where pressure P and liquid composition x are unknown. The calculation procedure is to input temperature and vapor composition and obtain pressure and the equilibrium liquid composition. The subroutines used are INPUT1, VIRIAL, PHIMIX, RSTATE and ACTCO.

The basic equation is the equation(I-B-1.4) of the previous chapter.

$$\Phi_i * y_i * P = \gamma_i * x_i * f_i^{0L} \qquad \text{(I-B-1.4)}$$

This equation can be transformed as below.

$$\Phi_i * y_i * P = \gamma_i * x_i * [f_i^{0L(P_r)} * \exp\{v_i^L * (P-P_r)/(R*T)\}] \qquad \text{(II-B-2.1)}$$

where γ_i is the activity coefficient, and the standard state fugacity, in brackets, is the product of the standard state fugacity at the reference pressure P_r and the Poynting correction. Equation (1) assumes that the molar volume v_i^L is a function of temperature only. Therefore, the quantity $f_i^{0L(P_r)}$, which is written $f_i^{(P_o)}$ if P_r=0, is the fugacity of pure liquid i at the system temperature, and the reference pressure. By using a reference pressure P_r=0, equation (II-B-2.1) is reduced to

$$\Phi_i*y_i*P=\gamma_i*x_i*[f_i^{(P_o)}*\exp\{v_i^L*P/(R*T)\}] \qquad \text{(II-B-2.2)}$$

with this equation P and x_i values are calculated at given y_i and temperature. All fixed parameter values are supplied by INPUT1, Φ_i is calculated by VIRIAL and PHIMIX, γ_i is supplied by ACTCO, and $f_i^{(P_o)}$ and v_i^L are calculated by RSTATE.

Rearrangement of equation(2) gives

$$x_i=[\Phi_i*y_i*P]/[\gamma_i*f_i^{(P_o)}*\exp\{v_i^L*P/(R*T)\}] \qquad \text{(II-B-2.3)}$$

The sum of all the x_i must equal unity, giving:

$$P^{-1}=\sum_{i=1}^{m}[\Phi_i*y_i]/[\gamma_i*f_i^{(P_o)}*\exp\{v_i^L*P/(R*T)\}] \quad \text{(II-B-2.4)}$$

Equation (4) may be solved iteratively for P; the x_i will be determined as well at each iteration by equation (3).

Figure II-B-2 gives the flow chart of the calculation procedure.

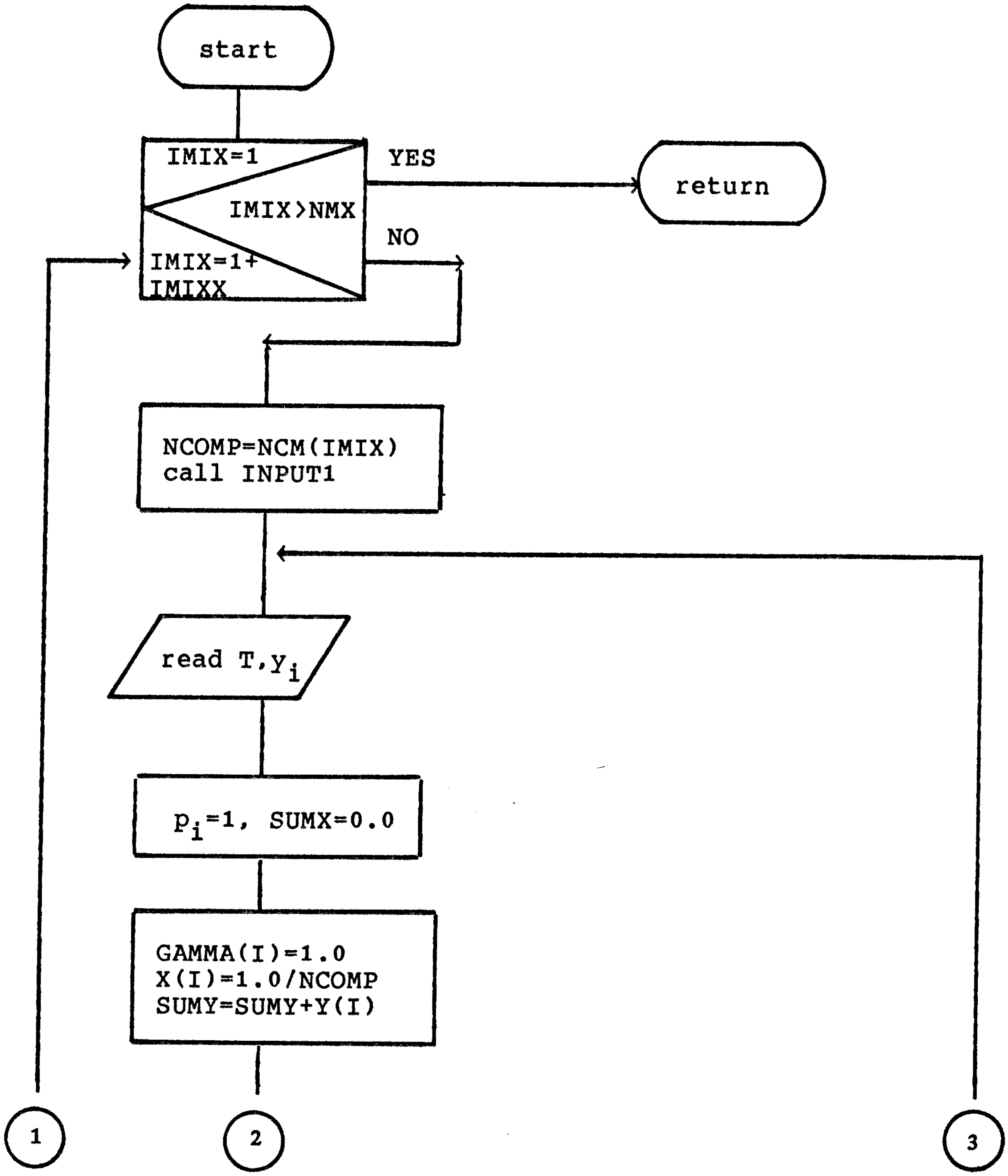

Figure II-B-2-a. Flow chart of the subroutine DEWP.

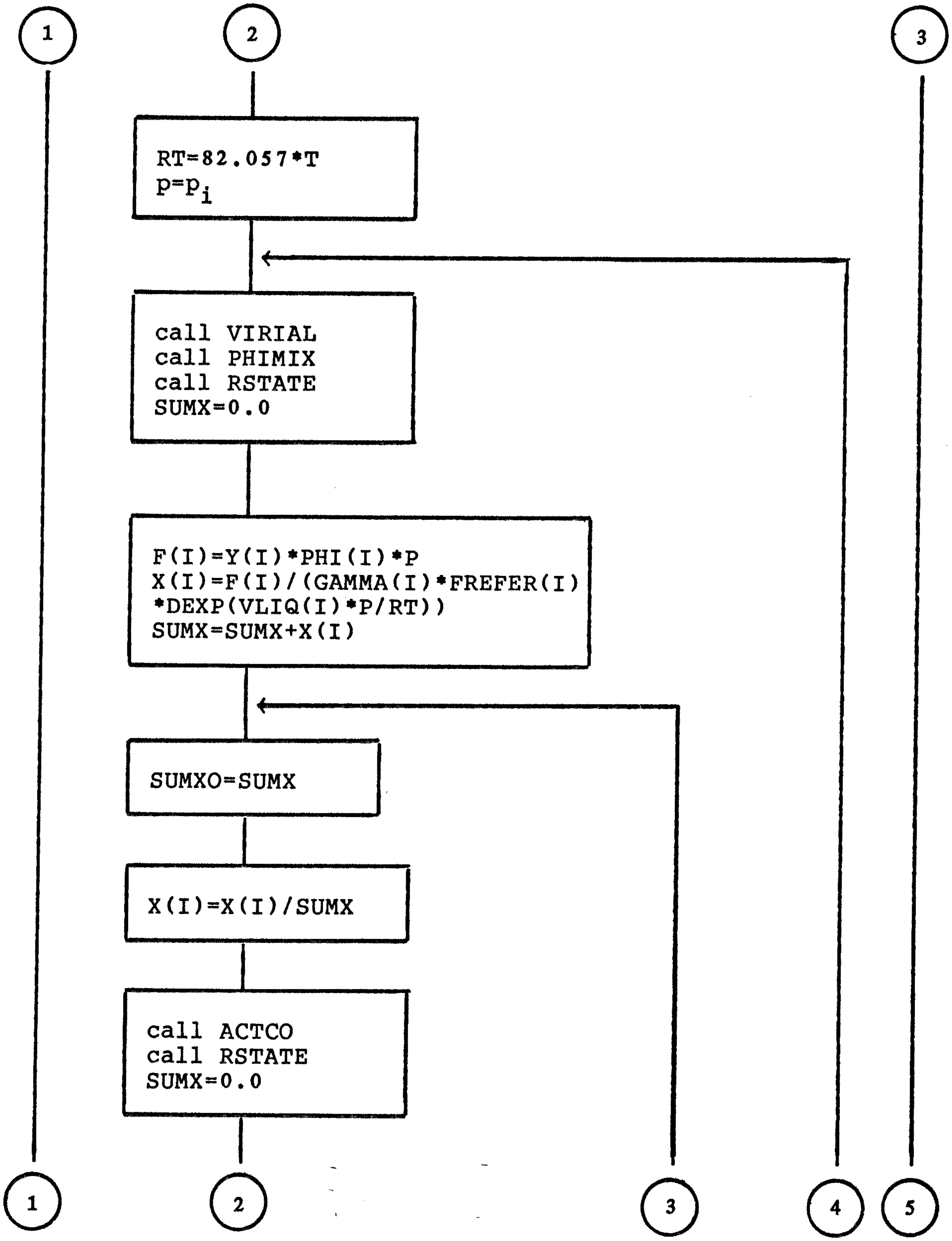

Figure II-B-2-b. Flow chart of the subroutine DEWP.

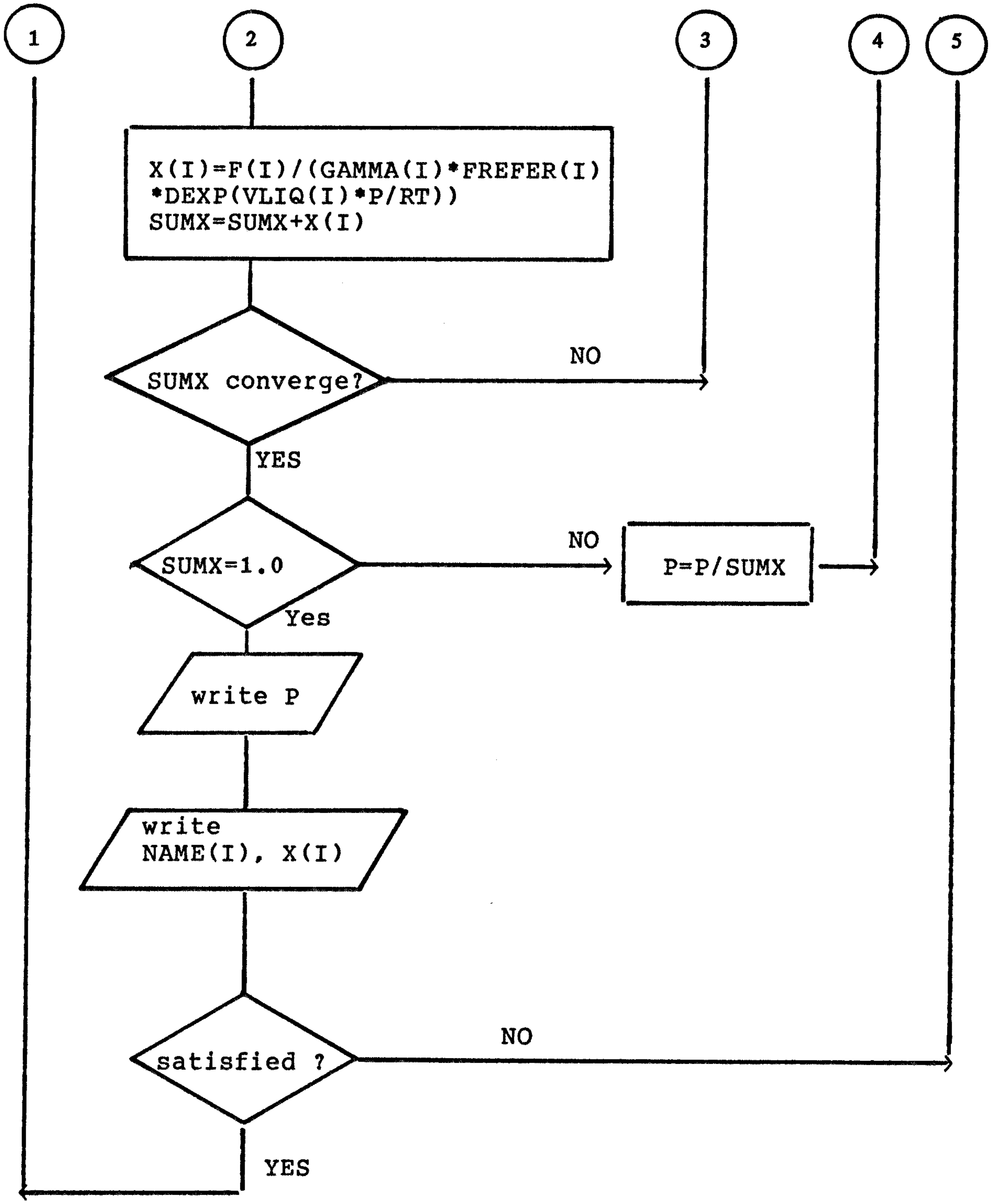

Figure II-B-2-c. Flow chart of the subroutine DEWP.

3. DEWT

Subroutine DEWT is the program for the calculation of the dewpoint temperature which represents a Vapor/Liquid Equilibrium calculation where Temperature T and liquid composition x are unknown. The calculational procedure is to input pressure and vapor composition and evaluate the corresponding temperature and liquid composition of the system. The subroutines used by this program are INPUT1, VIRIAL, PHIMIX, RSTATE, and ACTCO. The calculation procedure of dewpoint temperature is quite similar to DEWP, which was described in previous section, except that the roles of pressure and temperature are reversed.

The mathematical treatment is similar to DEWP in that it uses the same equation.

$$\Phi_i * y_i * P = \gamma_i * x_i * [f_i^{(P_o)} * \exp\{v_i^L * P/(R*T)\}] \qquad \text{(II-B-2.2)}$$

In this equation T and x_i values are calculated by given y_i and pressure. All fixed parameter values are supplied by INPUT1, The Φ_i value are calculated by VIRIAL and PHIMIX, γ_i is supplied by ACTCO and $f_i^{(P_o)}$, v_i^L are calculated by RSTATE.

As before, thus equation (II-B-2.2) can be rearranged to solve for x_i:

$$x_i=[\Phi_i*y_i*P]/[\gamma_i*f_i^{(P_o)}*\exp\{v_i^L*P/(R*T)\}] \quad \text{(II-B-2.3)}$$

and the x_i must sum to unity, giving:

$$1=\sum_{i=1}^{m}[\Phi_i*y_i*P]/[\gamma_i*f_i^{(P_o)}*\exp\{v_i^L*P/(R*T)\}] \quad \text{(II-B-3.1)}$$

thus equation must be solved by trial for T, which is implicit in the equation. As before, the x_i are calculated at each iteration by equation (II-B-2.3).

There are some difficulties with this procedure. One is that several variables are strongly dependent on temperature, so that the initial temperature guess is very important. If the initial guess is too far from the correct solution, later steps do not necessarily approach the solution. For this reason, determining the correct interaction parameter value to use to calculate the activity coefficients is very difficult, because the interaction parameter values are quite sensitive to temperature. The other problem is to set up an algorithm which can accomplish a rapid convergence. The present algorithm is inefficient in so far as the calculation time is concerned, although the method enables the program to obtain the correct solution. These problems should be addressed in next version of the program.

Figure II-B-3 gives the flow chart of the calculational procedures.

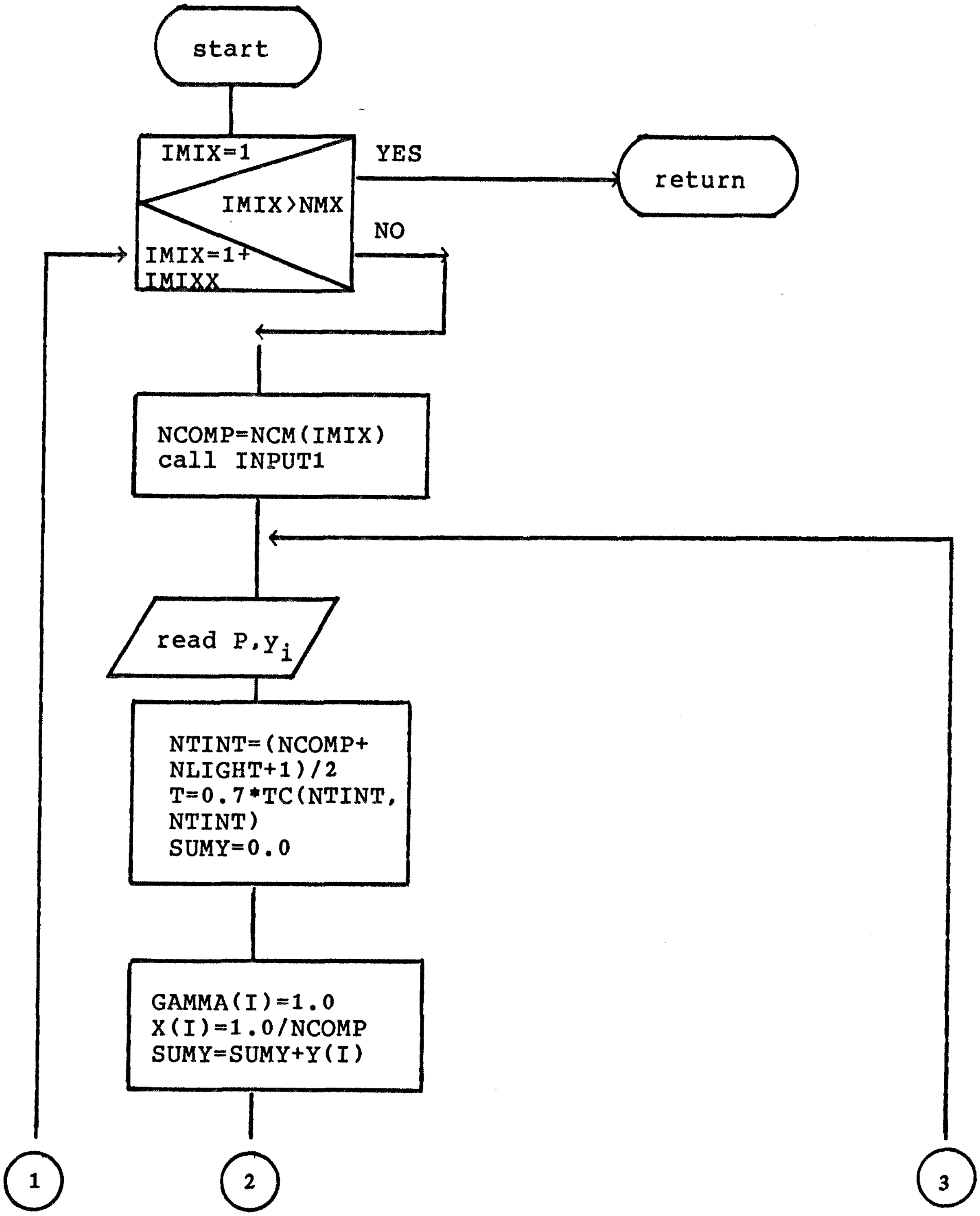

Figure II-B-3-a. Flow chart of the subroutine DEWT.

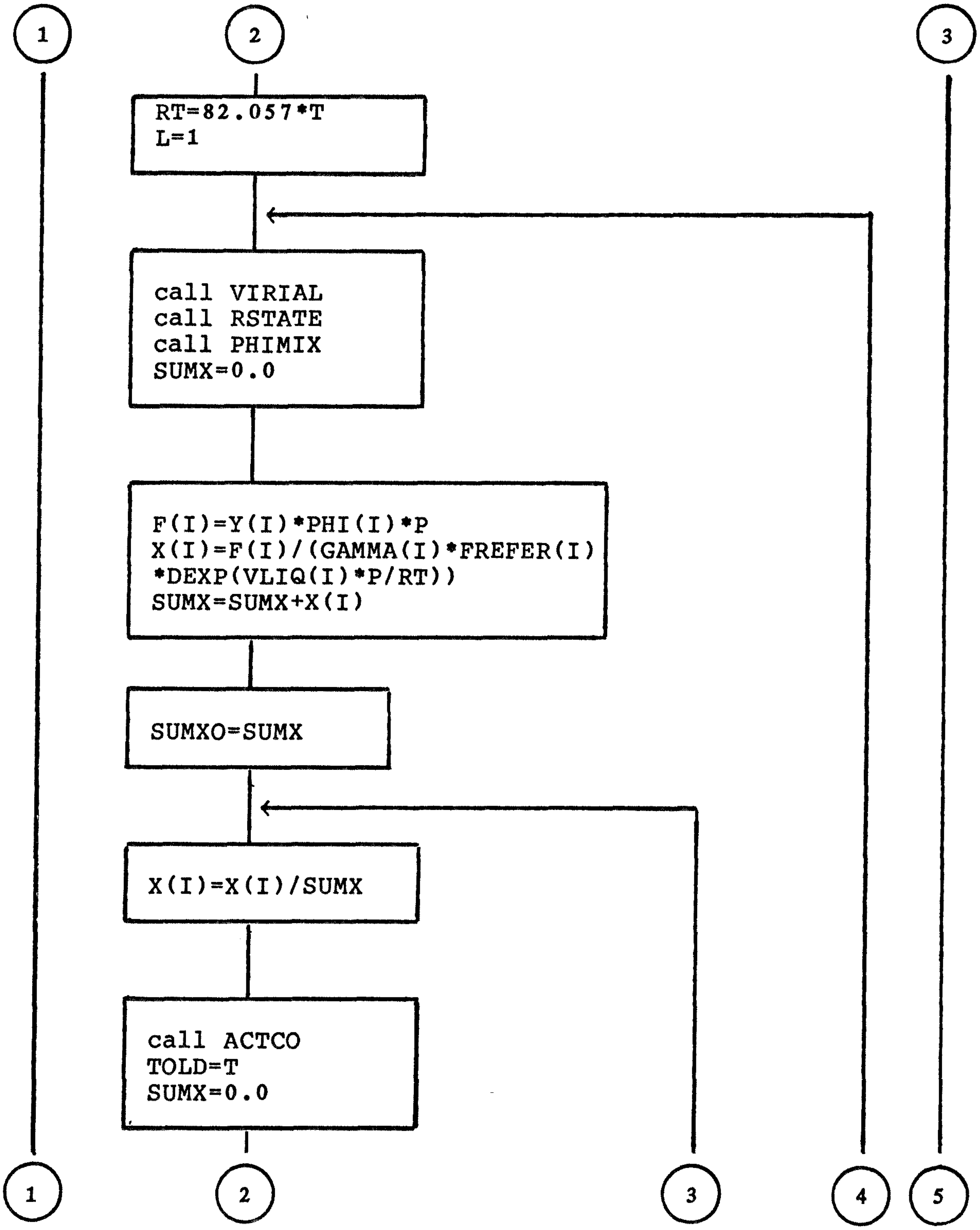

Figure II-B-3-b. Flow chart of the subroutine DEWT.

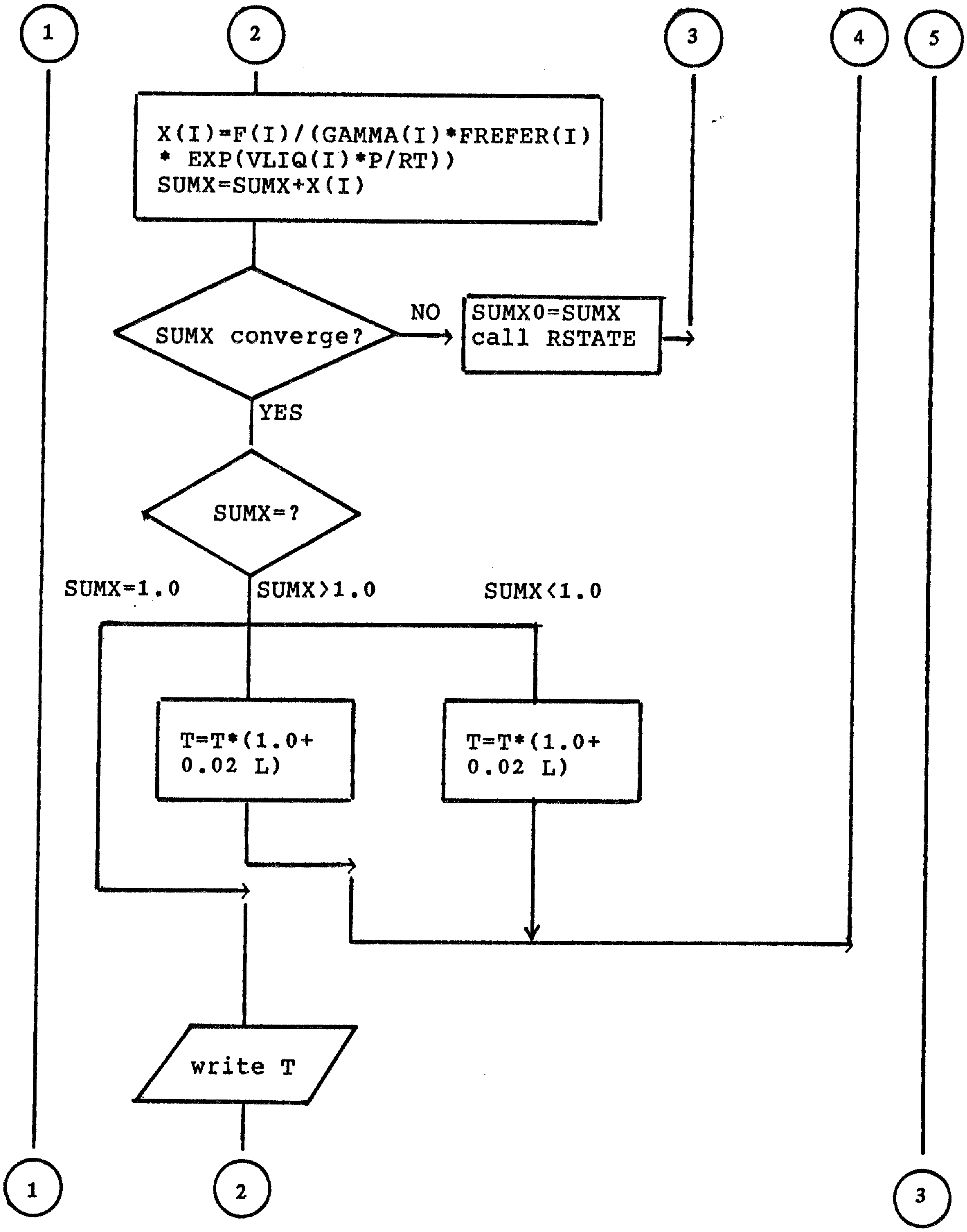

Figure II-B-3-c. Flow chart of the subroutine DEWT.

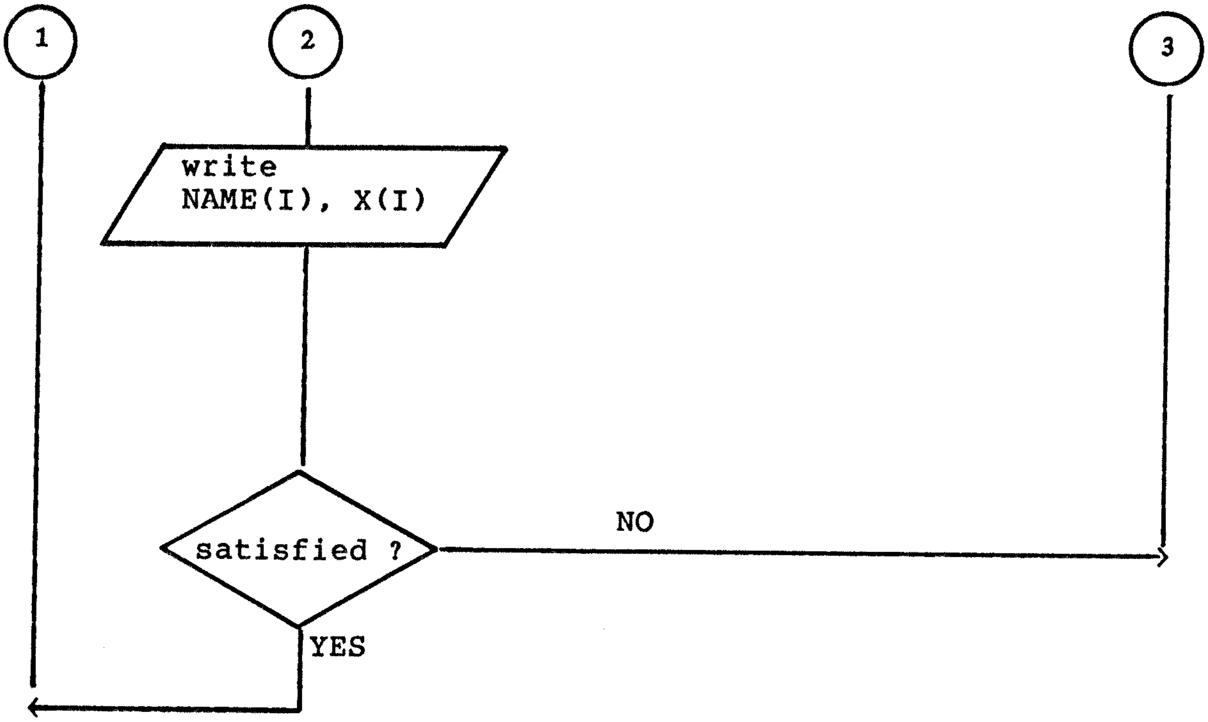

Figure II-B-3-d. Flow chart of the subroutine DEWT.

4. MIXHE

Subroutine MIXHE is the subroutine program for the calculation of the heat of mixing in the liquid phase. The subroutine reads liquid composition and temperature as input data interactively and gives the heat of mixing with units of either cal/mol or Joule/mol. The required subroutines are INPUT1, ACTCO, RSTATE. As is the same with other cases, INPUT1 supplies all fixed parameters and RSTATE supplies the values of liquid molar volumes which are used to calculate the activity coefficients by Wilson model. Finally ACTCO supplies calculated heat of mixing values. This program is relatively simple since many of the calculations for each model are done in ACTCO and its subroutines. Figure II-B-4 gives the flow chart of the program.

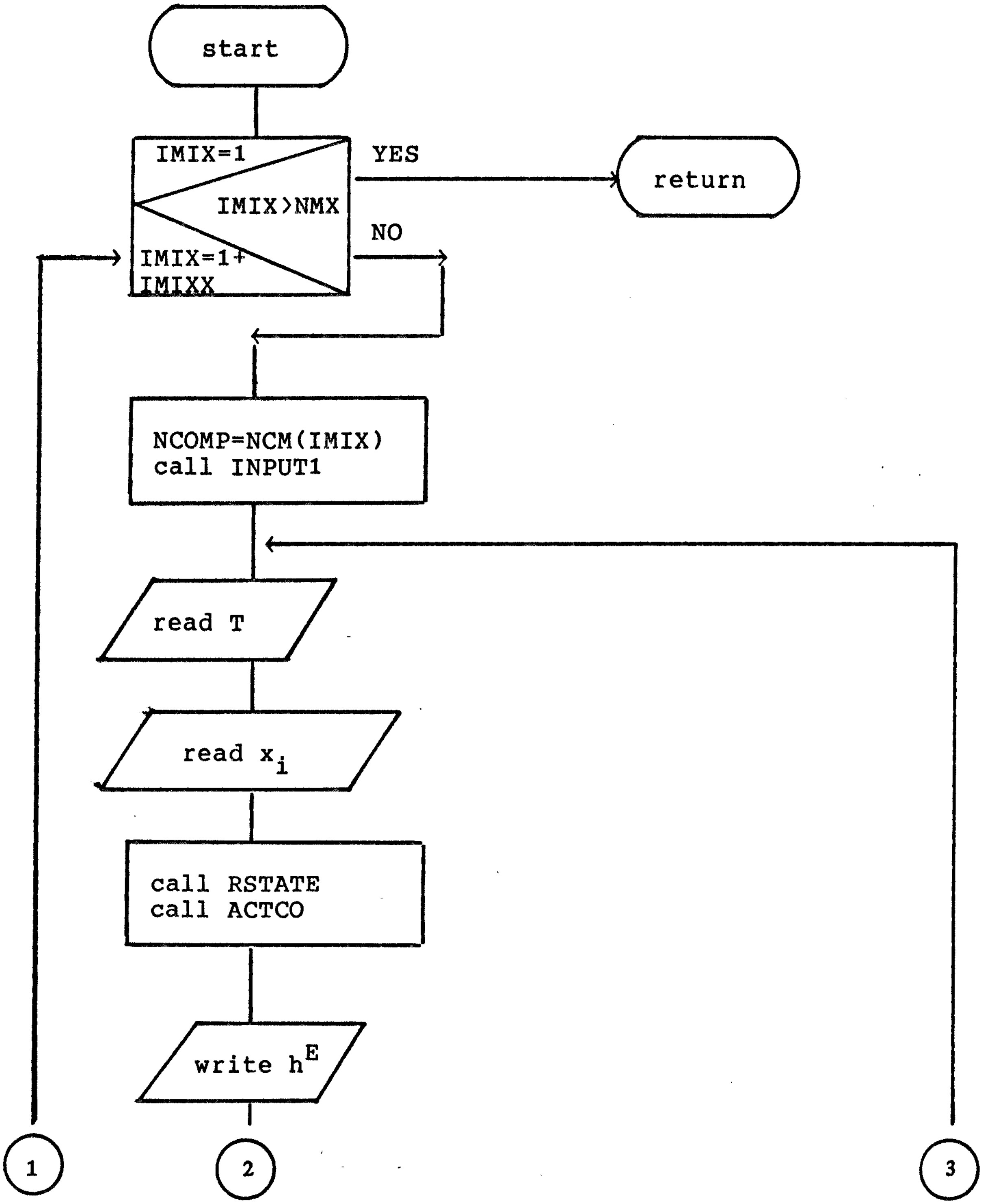

Figure II-B-4-a. Flow chart of the subroutine MIXHE.

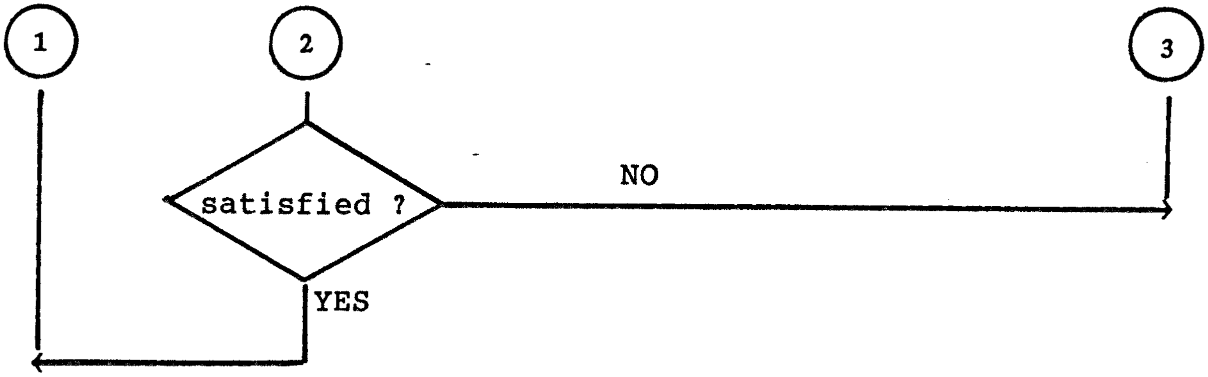

Figure II-B-4-b. Flow chart of the subroutine MIXHE.

5. ELIPS

Subroutine ELIPS is used for the purpose of calculating Liquid/Liquid two phase equilibrium separations. ELIPS uses the subroutines INPUT1, and LILIK. The input data are temperature, feed composition(Z_i) and the output consists of the compositions in both the extract phase(XE_i) and the raffinate phase(XR_i). The calculation procedure is similar to the isothermal Vapor/Liquid flash calculation. The basic mathematical expression to be used was mentioned in the introduction:

$$(\gamma_i * x_i)^{\alpha} = (\gamma_i * x_i)^{\beta} \qquad \text{(I-B-2.2)}$$

This can be rearranged by letting $K_i = \gamma_i^{\beta} / \gamma_i^{\alpha}$, $XE_i = x_i^{\alpha}$ and $XR_i = x_i^{\beta}$, thus:

$$XE_i = K_i * XR_i \qquad \text{(II-B-5.1)}$$

Another condition is the component material balances over the separation

$$F * Z_i = E * XE_i + R * XR_i \qquad \text{(II-B-5.2)}$$

$$F=E+R \qquad \text{(II-B-5.3)}$$

$$\sum_{i=1}^{m} XE_i=1, \quad \sum_{i=1}^{m} XR_i=1, \qquad \text{(II-B-5.4)}$$

By combining (1), (2), (3), (4) and setting $\alpha=E/F$, we obtain

$$XE_i=Z_i/\{(K_i-1)*\alpha+1\} \qquad \text{(II-B-5.5)}$$

$$XR_i=Z_i/\{(K_i-1)*\alpha+1\} \qquad \text{(II-B-5.6)}$$

By combining (4), (5) and (6) in the formulation of Rachford-Rice, we obtain

$$G(XR_i,\ XE_i,\ \alpha)=\sum_{i=1}^{m}[\{(K_i-1)*Z_i\}/\{(K_i-1)*\alpha+1\}]=0 \qquad \text{(II-B-5.7)}$$

The Z_i values are part of the input data, and the K_i values are calculated by subroutine LILIK. Equation (7) is solved by the Newton-Raphson method. Figure II-B-5 presents the flow chart of this calculation.

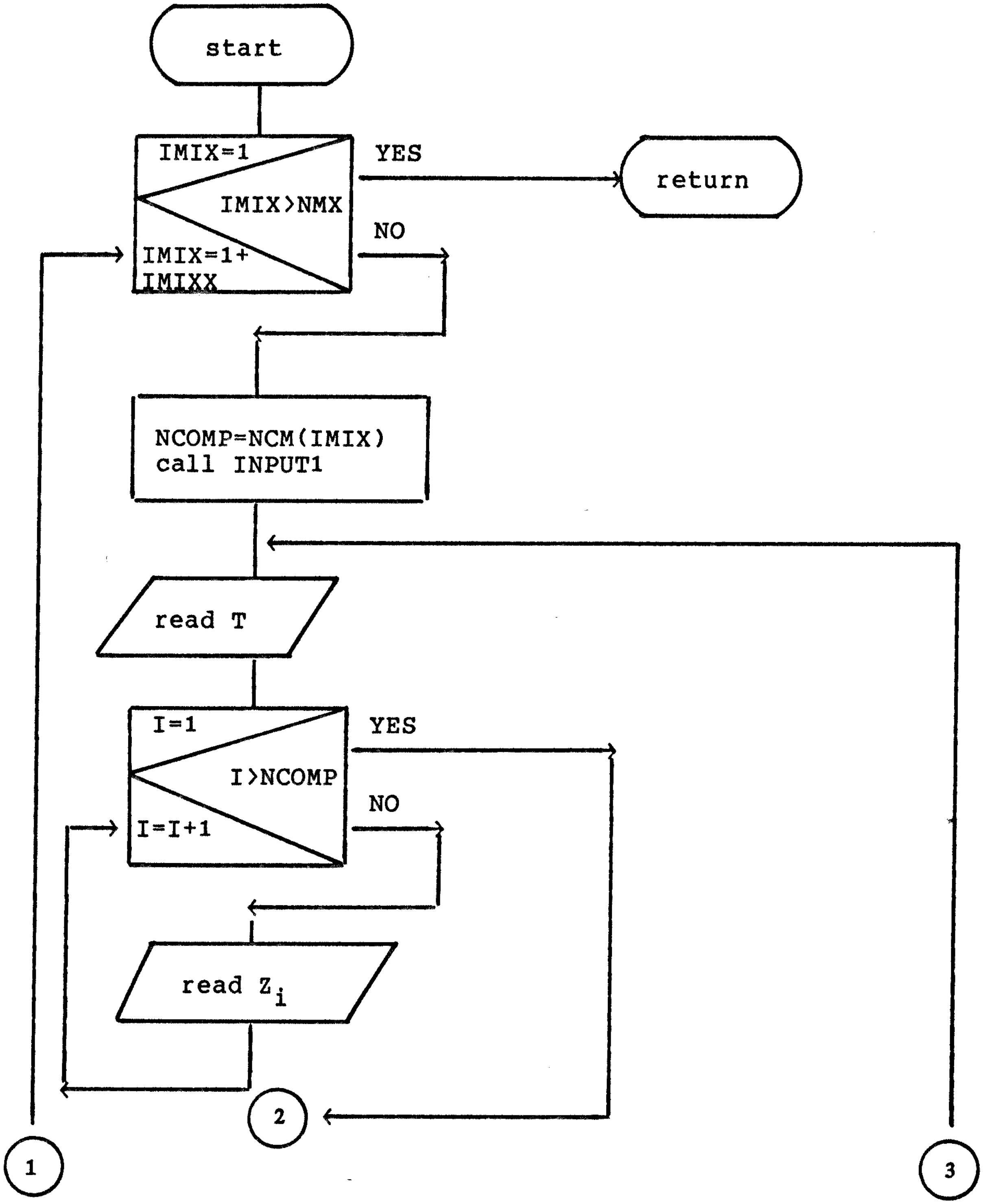

Figure II-B-5-a. Flow chart of the subroutine ELIPS.

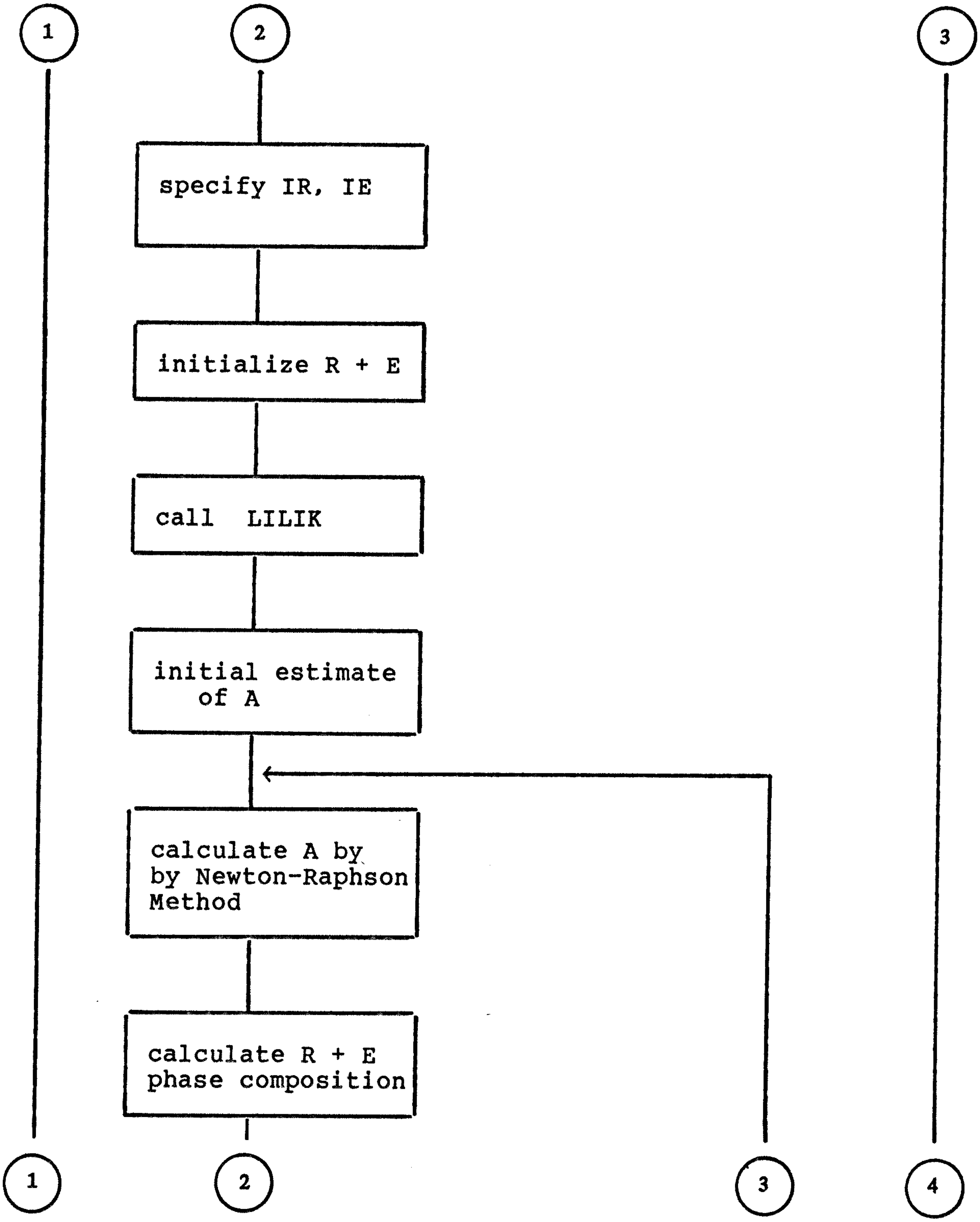
1
2
3
specify IR, IE
initialize R + E
call LILIK
initial estimate of A
calculate A by by Newton-Raphson Method
calculate R + E phase composition
1
2
3
4

Figure II-B-5-b. Flow chart of the subroutine ELIPS.

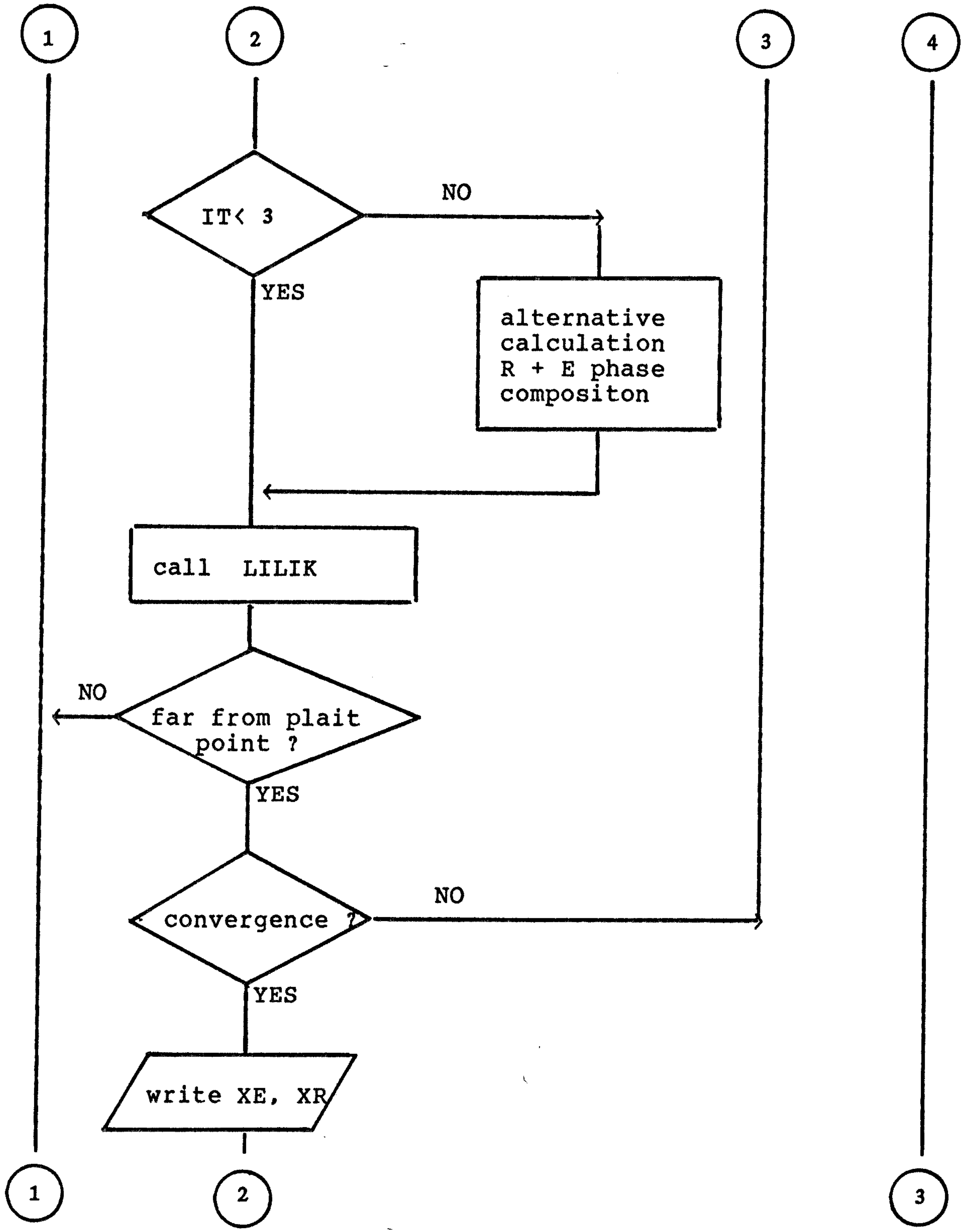

Figure II-B-5-c. Flow chart of the subroutine ELIPS.

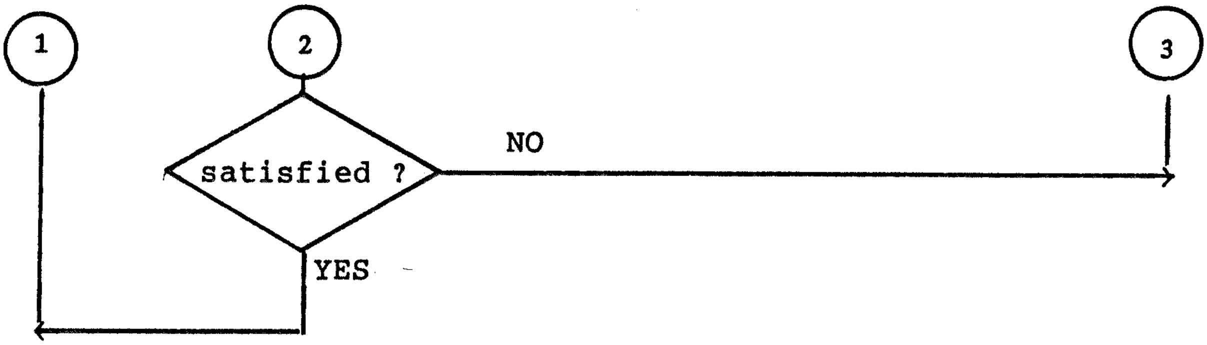

Figure II-B-5-d. Flow chart of the subroutine ELIPS.

6. LILIK

Subroutine LILIK is used for the purpose of calculating the equilibrium ratio K_i for liquid-liquid equilibrium. LILIK uses subroutine ACTCO, and input data being supplied automatically by the calling routine, ELIPS. Upon completion, the K_i values are transferred back to the calling routine.

The definition of K_i is:

$$K_i = \gamma_i^R / \gamma_i^E \qquad \text{(II-B-6.1)}$$

the needed γ_i values are supplied subroutine ACTCO.

Figure II-B-6 gives the subroutine flow chart.

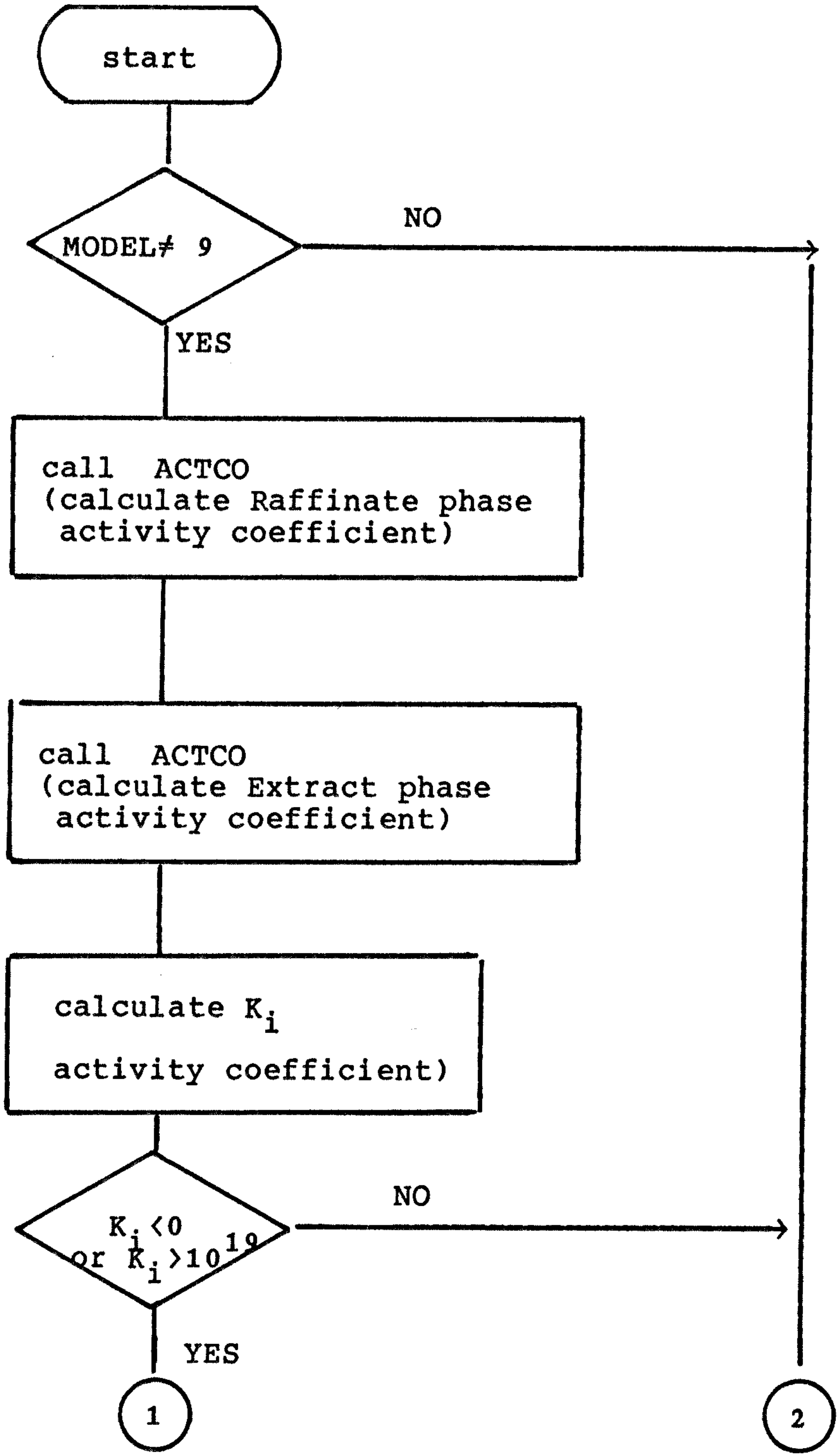

Figure II-B-6-a. Flow chart of the subroutine LILIK.

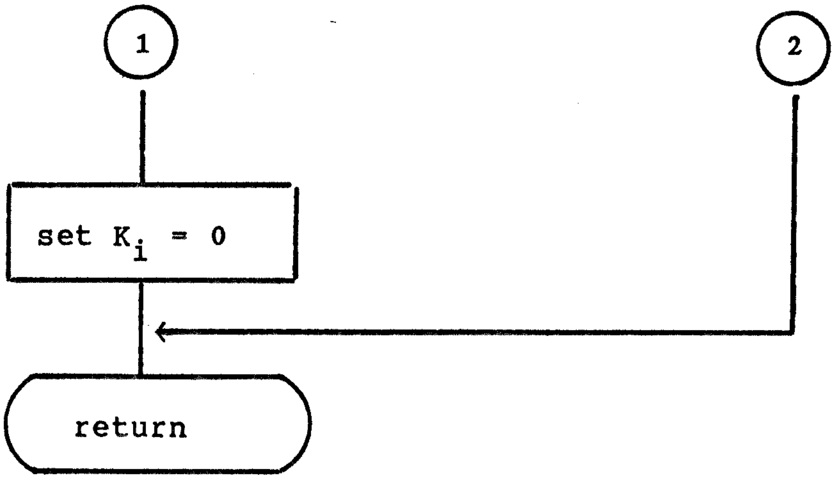

Figure II-B-6-b. Flow chart of the subroutine LILIK.

7. INPUT1

Subroutine INPUT1 is used for the purpose of supplying pure component properties and binary interaction parameters. It reads data from data files, keeps it in storage, conducts calculations for the mixture properties and trasfers these data to the other subroutines. Figure II-B-7 is the flow chart of subroutine INPUT1.

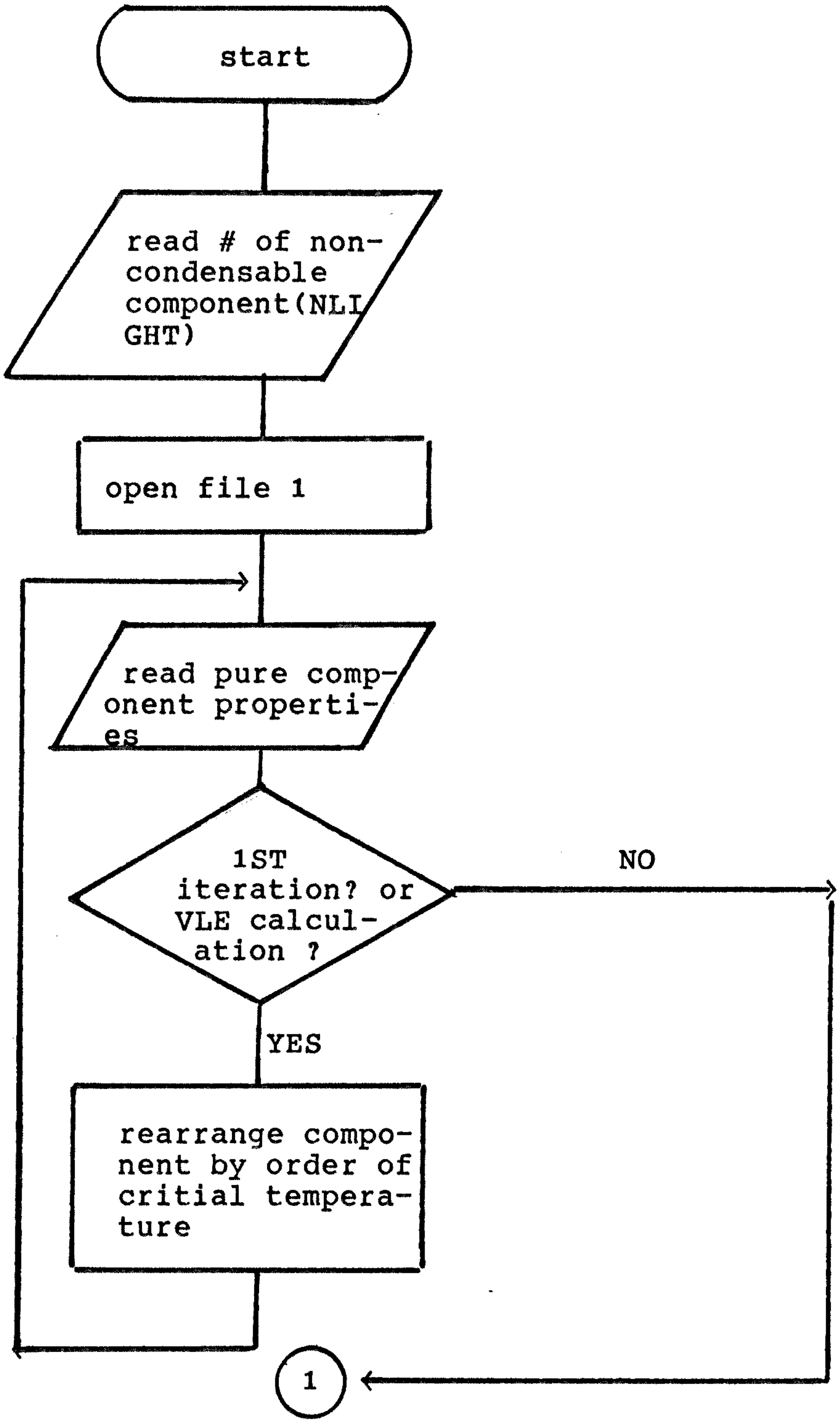
start
read # of non-condensable component(NLIGHT)
open file 1
read pure component properties
1ST iteration? or VLE calculation ?
NO
YES
rearrange component by order of critial temperature
1

Figure II-B-7-a. Flow chart of the subroutine INPUT1.

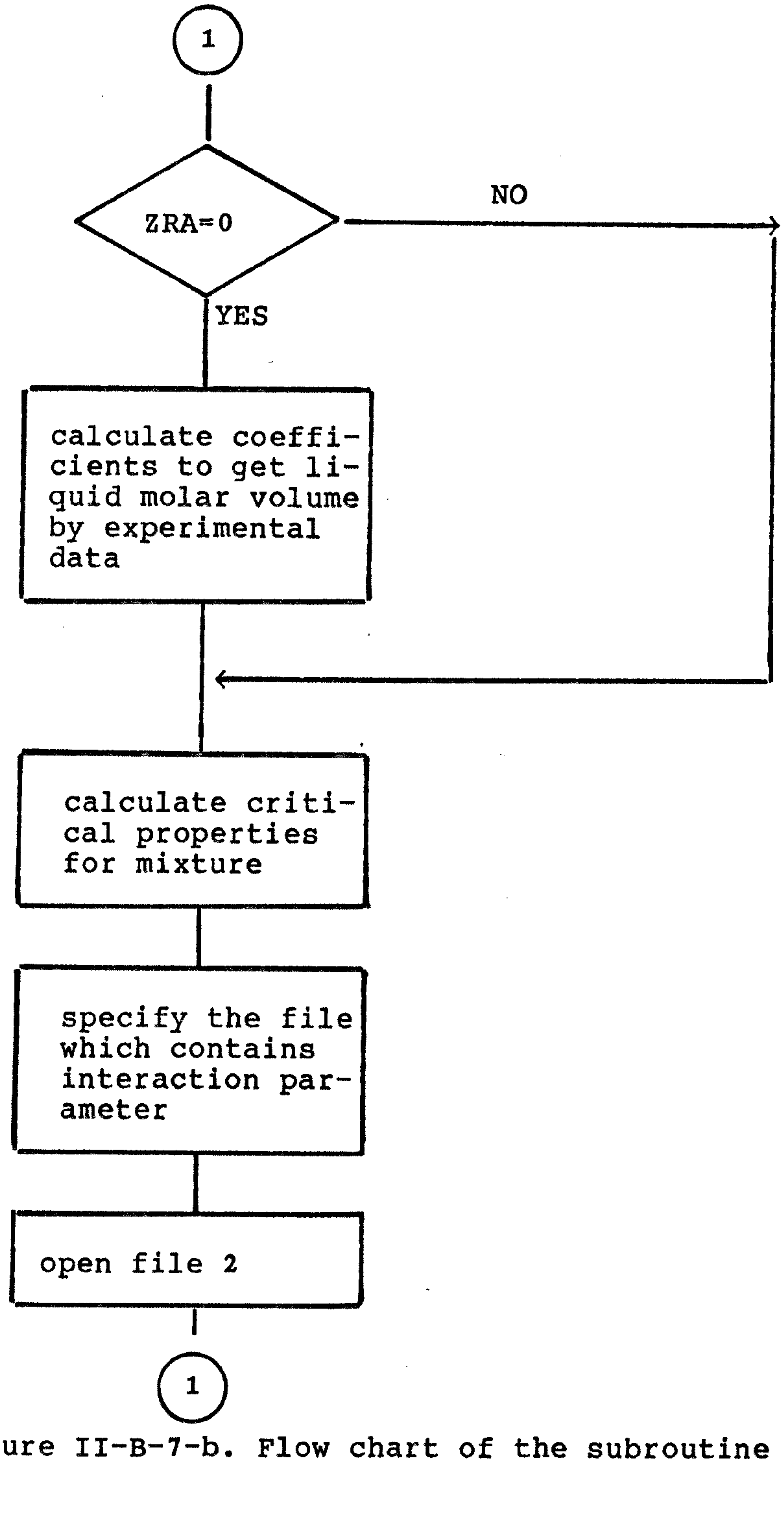
1
ZRA=0
NO
YES
calculate coefficients to get liquid molar volume by experimental data
calculate critical properties for mixture
specify the file which contains interaction parameter
open file 2
1

Figure II-B-7-b. Flow chart of the subroutine INPUT1.

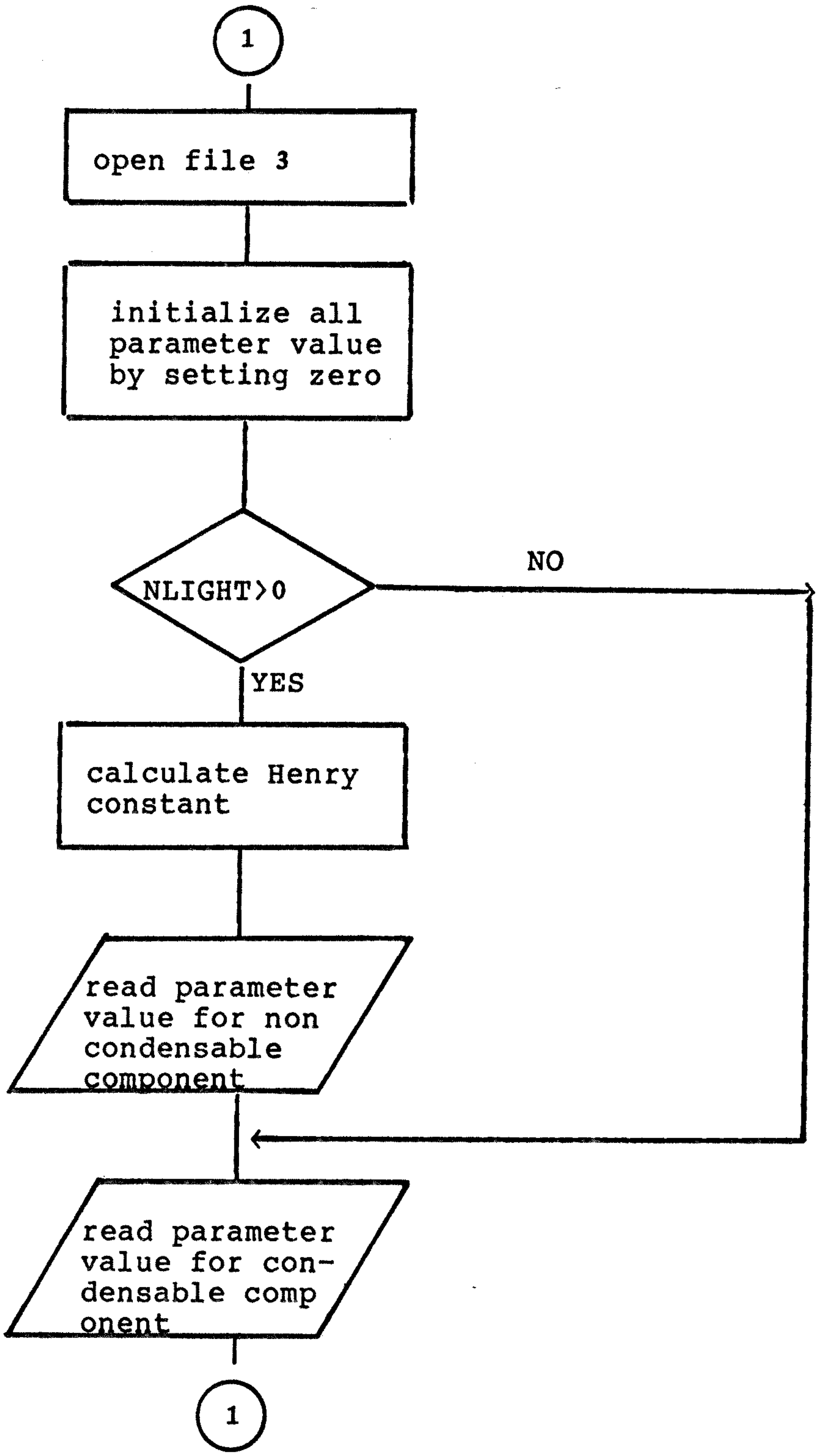
1
open file 3
initialize all parameter value by setting zero
NLIGHT>0
NO
YES
calculate Henry constant
read parameter value for non condensable component
read parameter value for con-densable comp onent
1

Figure II-B-7-c. Flow chart of the subroutine INPUT1.

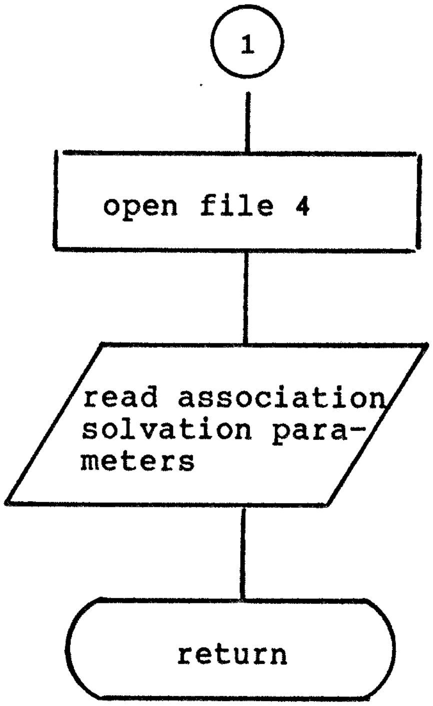

Figure II-B-7-d. Flow chart of the subroutine INPUT1.

8. VIRIAL

Subroutine VIRIAL calculates the second virial coefficients for the pure components and for all the binary pairs needed for the vapor -liquid equilibrium calculation. The second virial coefficients are calculated using the Hayden-O'Connell correlation(1). If the mixture includes organic acids, the equations of Hayden and O'Connell yield equilibrium constants for all the possible dimerization reactions.

In the generialized method, the pure-component and cross second virial coefficients B_{ij} are given by the sum of two contributions

$$B_{ij} = B^F{}_{ij} + B^D{}_{ij} \qquad \text{(II-B-8-1)}$$

where

$$B^F{}_{ij} = (B^F{}_{nonpolar})_{ij} + (B^F{}_{polar})_{ij} \qquad \text{(II-B-8-2)}$$

$$B^D{}_{ij} = (B_{metastable})_{ij} + (B_{bound})_{ij} + (B_{chemical})_{ij} \qquad \text{(II-B-8-3)}$$

Here, superscript F denotes relatively "free" molecules (weak physical forces), and D denotes relatively "bound" or "dimerized" molecules. The details of the calculations are given in the reference, and the schematic flow chart is given in Figure II-B-8.

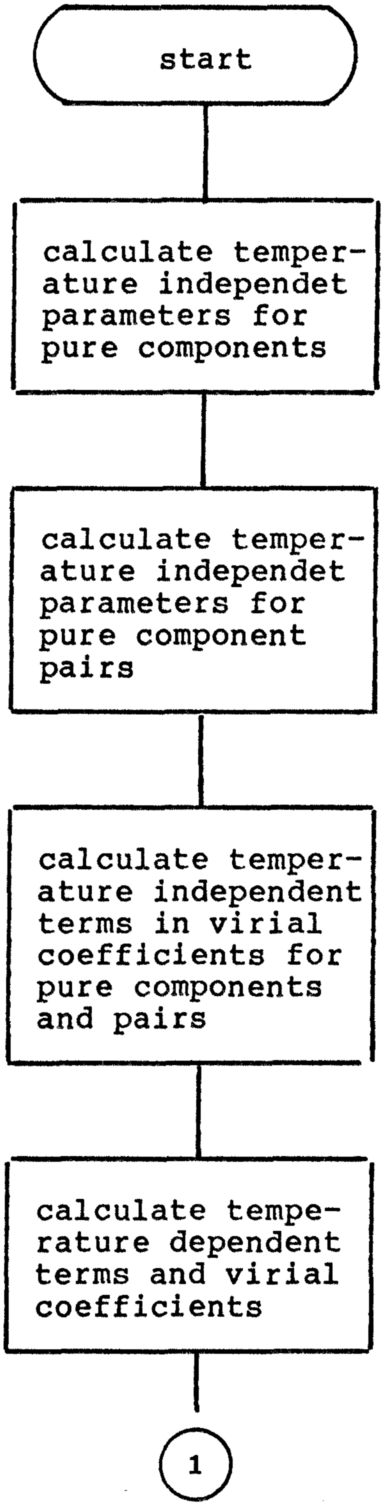

Figure II-B-8-a. Flow chart of the subroutine virial.

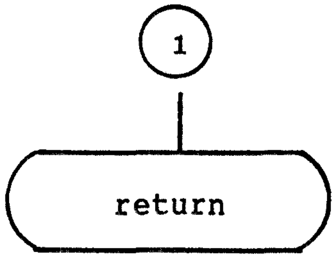

Figure II-B-8-b. Flow chart of the subroutine virial.

9. PHIMIX.

PHIMIX calculates vapor phase fugacity coefficients, Φ_i, for all components, and calls subroutine VIRIAL. The fugacity coefficient definition was given in the previous chapter.

$$\Phi_i = f_i^V/(P*y_i) \qquad \text{(I-B-1.2)}$$

Thus, it is a function of temperature, total pressure, and composition of the vapor phase. The fugacity coefficient can be found from an equation of state using the thermodynamic relation.

$$\ln(\Phi_i) = \{1/(R*T)\}* \int_0^P [(\partial v/\partial n_i)_{T,P,n\ j\neq i} - (R*T)/P]\ dP \qquad \text{(II-B-9.1)}$$

By introducing the virial equation, this can be transformed to:

$$\ln(\Phi_i) = (2*\sum_{j=1}^{m} y_j*\ B_{ij} - B_{mixture})*P/(R*T) \qquad \text{(II-B-9.2)}$$

where

$$B_{mixture}(T,\ y_1,\ y_2,\ \ldots,\ y_m) = \sum_{i=1}^{m}\ \sum_{j=1}^{m} y_i*y_j*B_{ij}(T) \qquad \text{(II-B-9.3)}$$

The, B_{ij} values are supplied by subroutine VIRIAL. The virial equation is appropriate for the describing deviations from ideality in those systems where moderate attractive forces yield fugacity coefficients not far from unity. However, in systems containing carboxylic acids, there prevails an entirely different physical situation; since two acid molecules tend to form a pair of stable hydrogen bonds, large negative deviations from vapor ideality occur even at very low pressures. To solve this problem, "chemical" theory(2) is introduced. The "chemical" theory postulates that there is a dimerization equilibrium of the type

$$i+j <=> ij$$

where i and j are two monomer molecules. In this case the fugacity coefficient of component i is given by

$$\Phi_i= Z_i*\Phi^{\neq}{}_i/y_i \qquad \text{(II-B-9.4)}$$

where Z is the "true" mole fraction; $\Phi^{\neq}$ is the fugacity coefficient of the true species. In here

$$\ln(\Phi^{\neq}{}_i)= B^F{}_i*P/(R*T) \qquad \text{(II-B-9.5)}$$

Z_i can be calculated by material balance and chemical equilibria. The flow chart is given in Figure II-B-9.

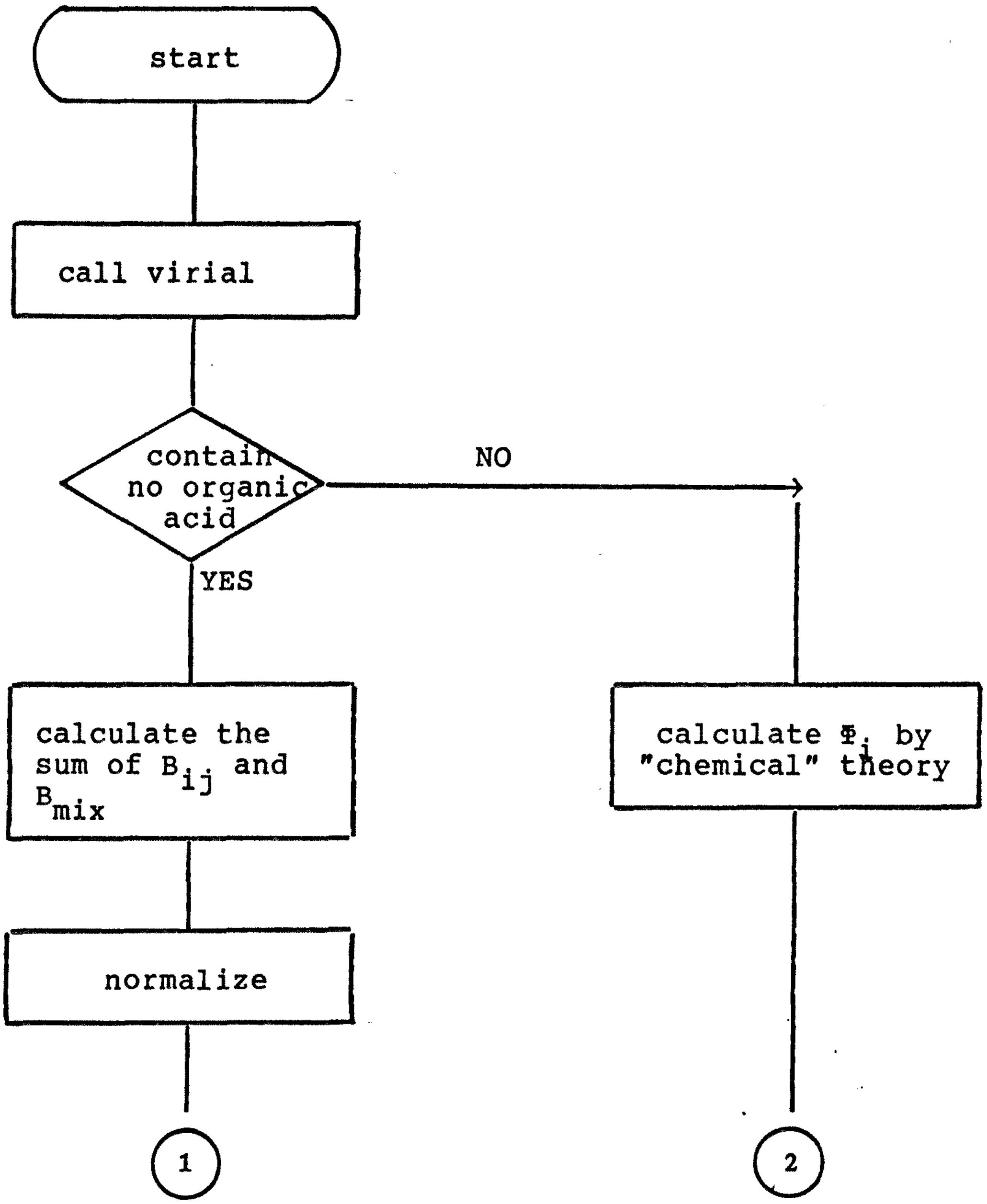
start
call virial
contain no organic acid
NO
YES
calculate the sum of B_{ij} and B_{mix}
normalize
calculate Φ_i by "chemical" theory
1
2

Figure II-B-9-a. Flow chart of the subroutine PHIMIX.

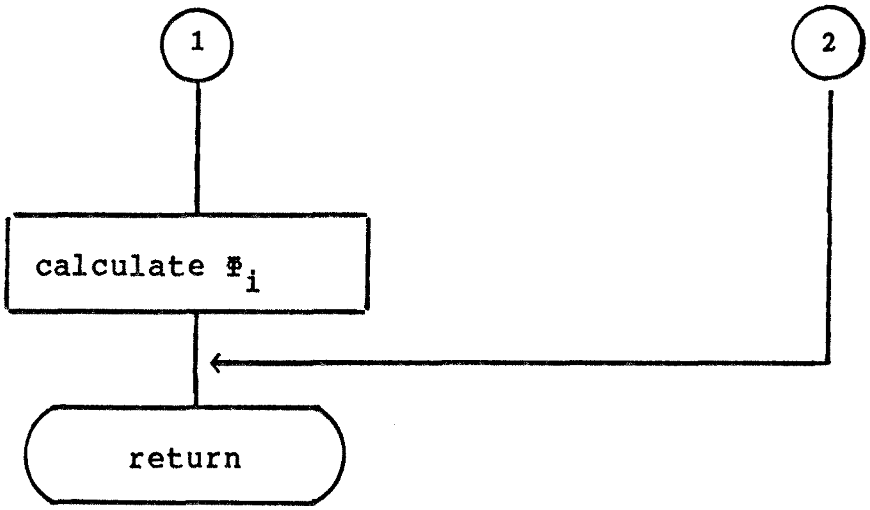

Figure II-B-9-b. Flow chart of the subroutine PHIMIX.

10. RSTATE

Subroutine RSTATE calculates the standard state properties of the pure liquids at the temperature in question. For subcritical components this is quite straightforward and depends only on temperature. However, if the system temperature is higher than the critical temperature of one or more species, calculation of the reference conditions becomes much more complex; furthermore, these quantities will be a function not only of temperature, but also of the other components present.

The liquid molar volume is expressed by reference(3) as

$$V^L_i = R*T_{ci}*(Z_{ri})^{\tau} /P_c \qquad \text{(II-B-10.1)}$$

where T_c, P_c are the critical temperature and pressure and Z_r is the modified Rackett parameter. τ is expressed in two cases.

$$\tau =1+(1-T_r)^{0.286} \qquad \text{for} \quad T_r \leq 0.75 \qquad \text{(II-B-10.2)}$$

$$\tau =1.6+0.00693/(T_r-0.655) \text{ for } T_r >0.75 \qquad \text{(II-B-10.3)}$$

if there is no available Z_r data, the volume must be calculated by regression of experimental data.

The vapor pressure is calculated by the Antoine equation.

$$P_{sat\ i} = EXP\{ A_i - B_i /(T+C_i)\} \qquad (II-B-10.4)$$

For the calculation of standard-state fugacity, some previous steps, as suggested in reference(4), are needed. This is a three parameter corresponding states correlation. Let

$$FSAT0F=-1.1970522/TR^2-1.3785023/TR+2.0048841 \qquad (II-B-10.5)$$

$$FSAT1F=-2.7741817/TR^2 +1.5454928/TR +1.3057555 \qquad (II-B-10.6)$$

$$PHIS0F=0.57335015/TR^3-3.076574/TR^2+5.6085595/TR-3.5021358 \qquad (II-B-10.7)$$

$$PHIS1F = 0.012089114/TR^{12}-.015172164/TR^{11}-0.068603516/TR^{10}+0.024364816/TR^9+0.14936909/TR^8+0.28917037/TR^7-0.12147436/TR^6-0.10665730/TR^5-1.1662283/TR^4+ 0.12666184/TR^3+0.3166137/TR^2+4.3538729/TR-3.7694018 \qquad (II-B-10.8)$$

and let PHIS=EXP(PHIS0F+ω*PHIS1F) in case of 1> TR > 0.56 and PHIS=EXP0{(B$_{ii}$*P$_{sat\ i}$/(R*T)} in case of TR $\leq$ 0.56, then standard-state fugacity, f_i^{0L}, is

$$f_i^{0L} = P_{sat\ i} * \text{PHIS}* \text{EXP}\{-V^L_i * P_{sat\ i}/(R*T)\} \quad \text{(II-B-10.9)}$$

If TR> 1.0 but close to 1.0, then

$$f_i^{0L} = P_{ci} * \text{EXP(FAST0F}+\omega*\text{FAST1F)} \quad \text{(II-B-10.11)}$$

If TR > 1.0 and far from 1.0, then the standard-state fugacity is Henry's constant. Figure II-B-10 gives the flow chart.

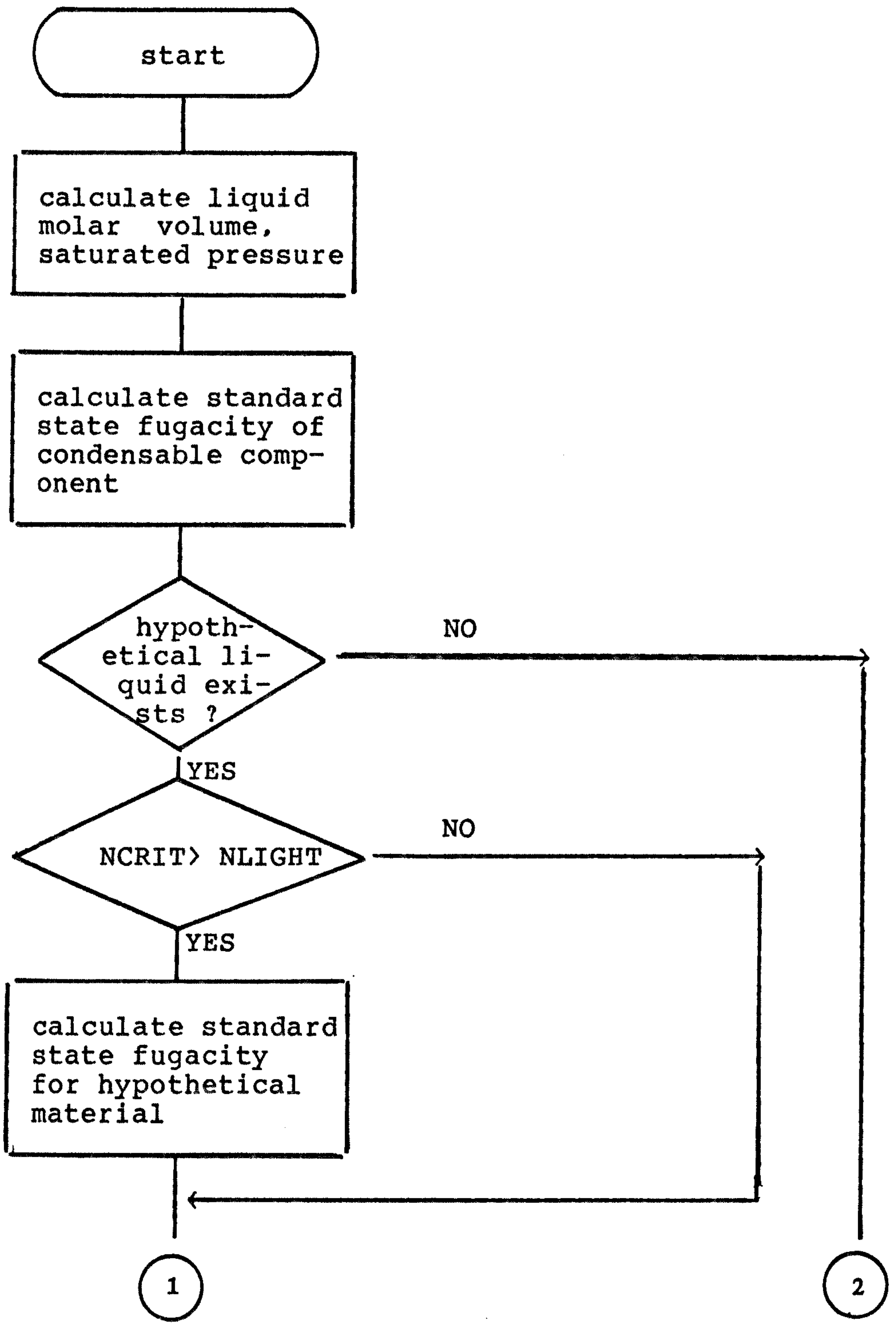
start
calculate liquid molar volume, saturated pressure
calculate standard state fugacity of condensable comp-onent
hypoth-etical li-quid exi-sts ?
NO
YES
NCRIT> NLIGHT
NO
YES
calculate standard state fugacity for hypothetical material
1
2

Figure II-B-10-a. Flow chart of the subroutine RSTATE.

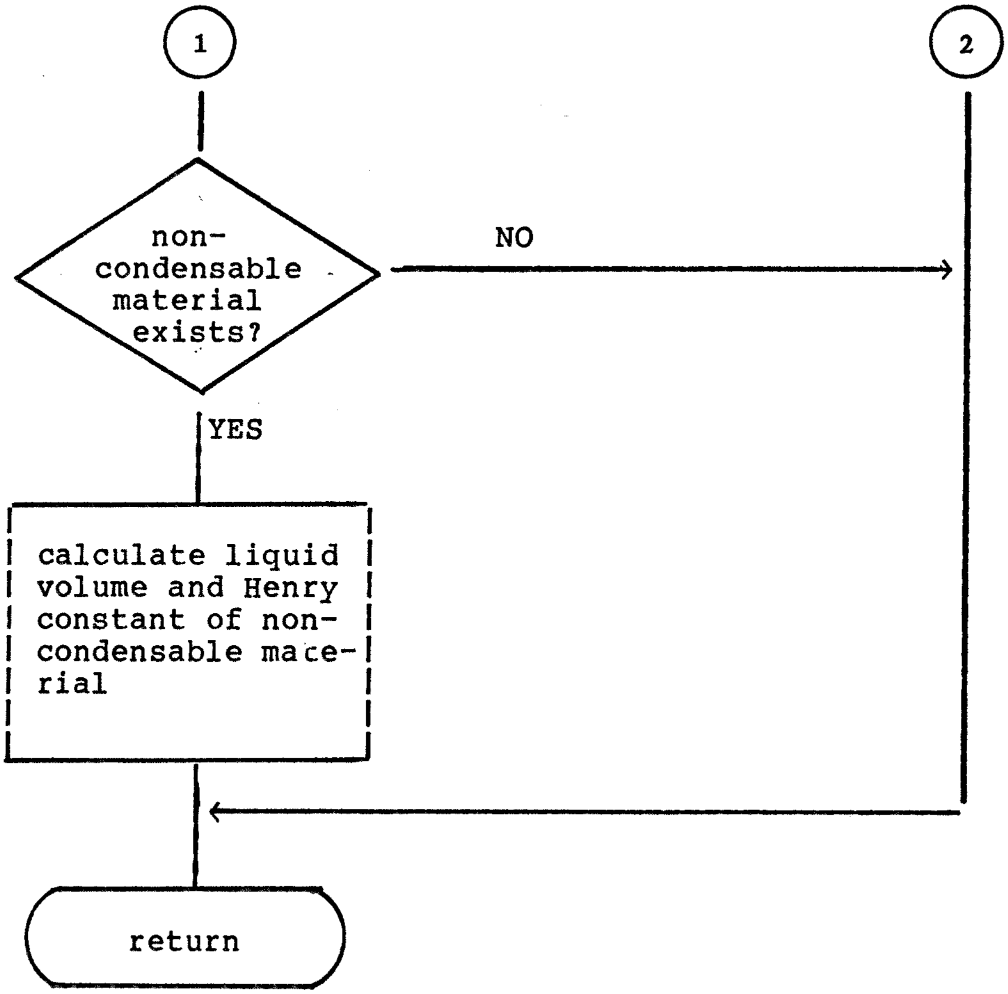
1
2
non-
condensable
material
exists?
NO
YES
calculate liquid volume and Henry constant of non-condensable macerial
return

Figure II-B-10-b. Flow chart of the subroutine RSTATE.

11. ACTCO

Subroutine ACTCO oversees the activity coefficient calculation by the designated activity coefficient model. It uses subroutines from ACTCO1 to ACTCO20. At present, ACTCO1, ACTCO5, ACTCO6, ACTCO8, ACTCO9, and ACTCO10 are available. They correspond to the KNOX, UNIQUAC, modified UNIQUAC, NRTL and Redlich-Kister equations. Figure II-B-11 is the flow chart.

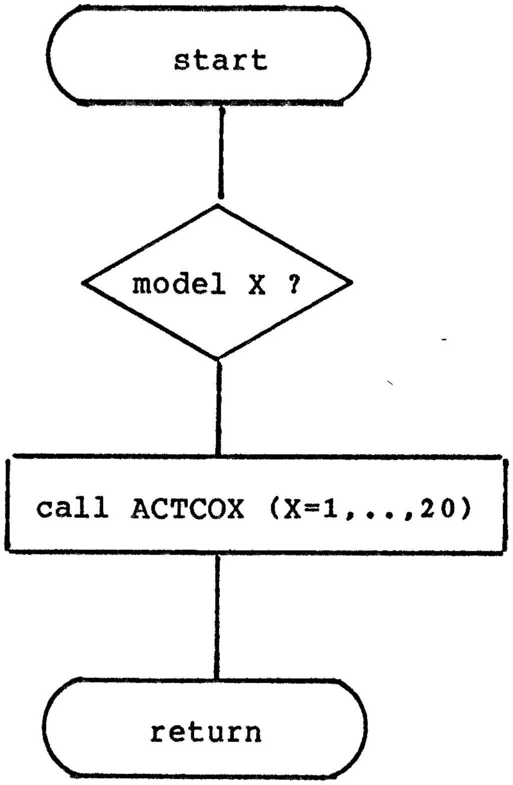

Figure II-B-11. Flow chart of the subroutine ACTCO.

12. ACTCO1

Subroutine ACTCO1 calculates activity coefficients and heat of mixing in liquid mixtures by the Knox Molecular (GAM) Model(5), and uses subroutine XSOLVE. This model is based on statistical arguments and developed as a local-composition model alternative to such two-fluid models as the UNIQUAC, NRTL and Wilson equation for the excess thermodynamic properties g^E and h^E.

The basic assumptions for the model are as follows:

1. The energies of molecular translation, rotation and vibration are unaffected by mixing; the energy change of mixing is therefore determined solely by the numbers and types of molecular interactions.
2. The interaction energy between a pair of molecules is constant, independent of temperature, pressure and composition.
3. Each molecule of species i engages in a characteristic number Z_i of pair interactions with other molecules.

The following are definitions for the final mathematical expression.

N_i : number of species-i molecules.

N_{ij} : number of i-j pair interactions.

x_i : mole fraction of species.

x_{ij} : local composition.

V_i : volume fraction.

θ_i : molecular Z fraction.

Φ_i : volume fraction.

ε_{ij} : interchange energy between species i and j.

$$\theta_i = \frac{Z_i*x_i/2}{\sum_{j=1}^{n} Z_j*x_j/2} \qquad \text{(II-B-12.1)}$$

$$x_{ij}= \frac{N_{ij}}{Z_i*N_i/2} \qquad \text{(II-B-12.2)}$$

$$C_{ij}=\mathrm{EXP}\left[\frac{-(2*\varepsilon_{ij}-\varepsilon_{ii}-\varepsilon_{jj})}{K*T}\right] \qquad \text{(II-B-12.3)}$$

$$\Phi_i= \frac{x_i*V_i}{\sum_{j=1}^{n} x_j*V_j} \qquad \text{(II-B-12.4)}$$

By these definitions excess properties and activity coefficients can be derived.

$$\frac{g^E}{R*T} = \sum_{i=1}^{n} x_i*\ln\frac{\Phi_i}{x_i} + \sum_{i=1}^{n} \frac{Z_i*x_i}{2} * \ln \frac{x_{ii}}{\Phi_i} \qquad \text{(II-B-12.5)}$$

$$\frac{h^E}{R*T} = \sum_{i=1}^{n} \sum_{j>i}^{n} \frac{-Z_i*x_i}{2} * x_{ij}*\ln C_{ij} \qquad \text{(II-B-12.6)}$$

$$\ln(\gamma_i)=1-\frac{\Phi_i}{x_i}+\ln\frac{\Phi_i}{x_i}+\frac{Z_i}{2}\left(\frac{\Phi_i}{\Theta_i}-1+\ln\frac{x_{ii}}{\Phi_i}\right) \quad \text{(II-B-12.7)}$$

From the empirical point of view it fits data for binary systems, both VLE and H^E data, significantly better than other models, and is at least comparable in other respects. From the theoretical point of view, it avoids the concept of hypothetical fluids while rigorously satisfying all material-balance constraints, thus suggesting a rational approach to the understanding of liquid mixtures from the molecular point of view.

13. XSOLVE

Subroutine XSOLVE calculates local composition x_{ij} by the fixed point iteration method. This subroutine is an auxiliary subroutine of subroutine ACTCO1.

14. ACTCO5

Subroutine ACTCO5 calculates liquid phase activity coefficients, GAMMA, and heat of mixing at given temperature T and liquid composition x by using the UNIQUAC model(6) and modified UNIQUAC model(7). The modified UNIQUAC model is slightly different from the UNIQUAC model; this difference is described below.

The excess Gibbs energy as described by either the UNIQUAC or the modified UNIQUAC model consists of two parts.

$$g^E = g^E(\text{combinatorial}) + g^E(\text{residual}) \qquad \text{(II-B-14.1)}$$

$$g^E(\text{combinatorial})/(R*T)=\sum_{i=1}^{n} x_i * \ln(\Phi_i/x_i) + (Z/2)*\sum_{i=1}^{n} q_i * x_i *\ln(\theta_i/\Phi_i) \qquad \text{(II-B-14.2)}$$

$$g^E(\text{residual})/(R*T)=-\sum_{i=1}^{n} q'_i * x_i*\ln(\sum_{j=1}^{n} \theta'_j*\tau_{ji}) \qquad \text{(II-B-14.3)}$$

where

$$\Phi_i=r_i*x_i/(\sum_{j=1}^{n} r_j*x_j),\ \theta_i= q_i*x_i/(\sum_{j=1}^{n} q_j*x_j) \text{ and}$$

$$\theta'_i= q'_i*x_i/(\sum_{j=1}^{n} q'_j*x_j).$$

Here, τ is the UNIQUAC parameter, Φ is the segment fraction, θ and θ' are area fractions, r, q and q'are pure-component molecular structure constants and Z equals 10. In the UNIUAC model q=q', while in the modified UNIQUAC model these are different for some specific components. For any component i, the activity coefficient is given by

$$\ln(\gamma_i) = \ln(\Phi_i/x_i) + (Z/2)*q_i*\ln(\theta_i/\Phi_i) + l_i - (\Phi_i/x_i)* \sum_{j=1}^{n} x_j*l_j - q'_i*\ln(\sum_{j=1}^{n} \theta'_j*\tau_{ji}) + q'_i - q'_i* \sum_{j=1}^{n} \frac{\theta'_j*\tau_{ij}}{\sum_{j=1}^{n} \theta'_j*\tau_{ij}} \qquad \text{(II-B-14.4)}$$

where

$$l_j=(Z/2)*(r_j-q_j)-(r_j-1)$$

and the heat of mixing is given by:

$$h^E = R* \sum_{i=1}^{m} \left[\frac{q'_i* x_i}{\sum_{j=1}^{m} \theta'_j*\tau_{ji}} * (\sum_{j=1}^{m} \theta'_j*\tau_{ji}*a_{ji}) \right] \qquad \text{(II-B-14.5)}$$

where a_{ij} is an energy parameter.

The major charactistics of the UNIQUAC equations are:

1. Applicability to multicomponent mixtures in terms of binary parameters only.
2. Applicability to liquid-liquid equilibria.
3. A built-in temperature dependence valid over at least moderate range.
4. Possibily superior representation of mixtures of widely different molecular sizes.

15. ACTCO8

Subroutine ACTCO8 calculates liquid phase activity coefficients(GAMMA) and heat of mixing at a given temperature T and liquid composition x_i by using the NRTL model(8) which uses a third parameter, α, to account for the non-randomness of the liquid solutions. The mathematical expression is

$$\ln(\gamma_i) = \frac{\sum_{j=1}^{m} \tau_{ji} * G_{ji} * x_j}{\sum_{l=1}^{m} G_{li} * x_l} + \sum_{j=1}^{m} \frac{x_j * G_{ij}}{\sum_{j=1}^{m} G_{lj} * x_l} * \left[\tau_{ij} - \frac{\sum_{n=1}^{m} x_n * \tau_{nj} * G_{nj}}{\sum_{l=1}^{m} G_{lj} * x_l} \right] \qquad \text{(II-B-15.1)}$$

where

G_{ij}=EXP($-\alpha_{ij}*\tau_{ij}$), $\tau_{ij}=\Delta g_{ij}/(R*T)$ and g is the residual Gibbs free energy. And h^E can be expressed as below

$$h^E = \sum_{i=1}^{m} \sum_{j=1}^{m} x_i * \frac{x_j * G_{ij}}{\sum_{j=1}^{m} x_j * G_{ij}} * \left(1 + \frac{x_i * \alpha_{ij} * \tau_{ij}}{\sum_{j=1}^{m} x_j * G_{ij}}\right) * R * T \qquad \text{(II-B-15.2)}$$

This model usually represents binary equilibrium data quite well with its three parameters. It is superior to the Wilson equation in that it can represent liquid-liquid equilibria. The principal shortcoming of the NRTL equation is the ambiguity of the parameter α.

16. ACTCO9

Subroutine ACTCO9 calculates liquid phase activity coefficients and heats of mixing at a given temperature T and liquid composition x_i by using the Wilson model(11). This model was the first one to use the local composition concept. The mathematical expressions are as below.

$$\ln(\gamma_i) = 1-\ln(\sum_{j=1}^{n} x_i*\Lambda_{ij}) - \sum_{k=1}^{n} \frac{x_k*\Lambda_{ki}}{\sum_{j=1}^{n} x_j*\Lambda_{kj}} \qquad \text{(II-B-16.1)}$$

where

$$\Lambda_{kj} = \frac{v_j}{v_k} * \mathrm{EXP}(\frac{-\Delta\lambda_{kj}}{R*T})$$

here v is the molar liquid volume and $\Delta\lambda_{kj}$ is the energy parameter value difference. The h^E expression is:

$$h^E = \sum_{i=1}^{m} \sum_{j=1}^{m} x_i* (\frac{x_j* \Lambda_{ij}}{\sum_{j=1}^{m} x_j*\Lambda_{ij}})* \Delta\lambda_{ij} *R*T \qquad \text{(II-B-16.2)}$$

This model shows excellent prediction ability for vapor-liquid equilibrium calculations involving hydrocarbons, alcohlos, ethers, esters, ketones and water. However, the Wilson equation can not predict phase splitting, and thus it can not be applied to liquid-liquid equilibrium calculations.

17. ACTCO10

Subroutine ACTCO10 calculates liquid phase activity coefficients and heats of mixing at a given temperature T and liquid composition x_i, using the Redlich-Kister equation(10). This subroutine can only be applied to binary systems. In the Redlich-Kister equation the excess Gibbs energy is expressed

$$g^E/(R*T)= x_1x_2*\{B+C*(x_1-x_2)+D*(x_1-x_2)^2+E*(x_1-x_2)^3 +F*(x_1-x_2)^4+G*(x_1-x_2)^5\} \qquad \text{(II-B-17.1)}$$

Generally

$$h^E= -T^2*\{(\partial g^E/T)/\partial T\}_{P,x} \qquad \text{(II-B-17.2)}$$

In this case, $g^E/(R*T)$ has no explicit temperature dependence in equation(II-B-17.1), and thus it can be treated as a constant with respect to temperature. Application of equation(2) to equation(1) thus results in:

$$h^E=G^E \qquad \text{(II-B-17.3)}$$

In here, let $P= B+C*(x_1-x_2)+D*(x_1-x_2)^2+E*(x_1-x_2)^3+F*(x_1-x_2)^4+G*(x_1-x_2)^5$ and $Q= C+2*D*(x_1-x_2)+3*E*(x_1-x_2)^2+4*F*(x_1-x_2)^3+5*G*(x_1-x_2)^4$. The activity coefficients expressions then become:

$$\gamma_1=EXP(x_2^2*P+2.0*x_1x_2^2*Q) \qquad \text{(II-B-17.4)}$$

$$\gamma_2=EXP(x_1^2*P-2.0*x_1^2x_2*Q) \qquad \text{(II-B-17.5)}$$

The Redlich-Kister equation is very simple but needs too many parameters. Furthermore, the coefficients B, C, D, E, F and G are strongly dependendent on temperature, with the result that the applicable temperature range is relatively narrow.

18. PARA

The role of subroutine PARA is to select the appropriate parameter control subroutine. This program uses 6 subroutines. They are GENCRE, GENINP, GENUP, PARACRE, PARAINP and PARAUP.

GENCRE is used for the creation of file named GEN.DAT which will contain the pure material properties. GENINP is used for mass data input to the file GEN.DAT. GENUP is used for the correction of data which are contained in GEN.DAT. PARACRE is used for the creation of files which will contain binary interaction parameters. The filenames are ASPARA.DAT, PARA1.DAT, PARA5.DAT, PARA8.DAT, PARA9.DAT and PARA10.DAT. PARAINP has a similar function to GENINP but is used for the files which are created by PARACRE. The routine PARAUP is similar to GENUP and is used for the files which are created by PARACRE. Figure II-B-12 is a flow chart for subroutine PARA.

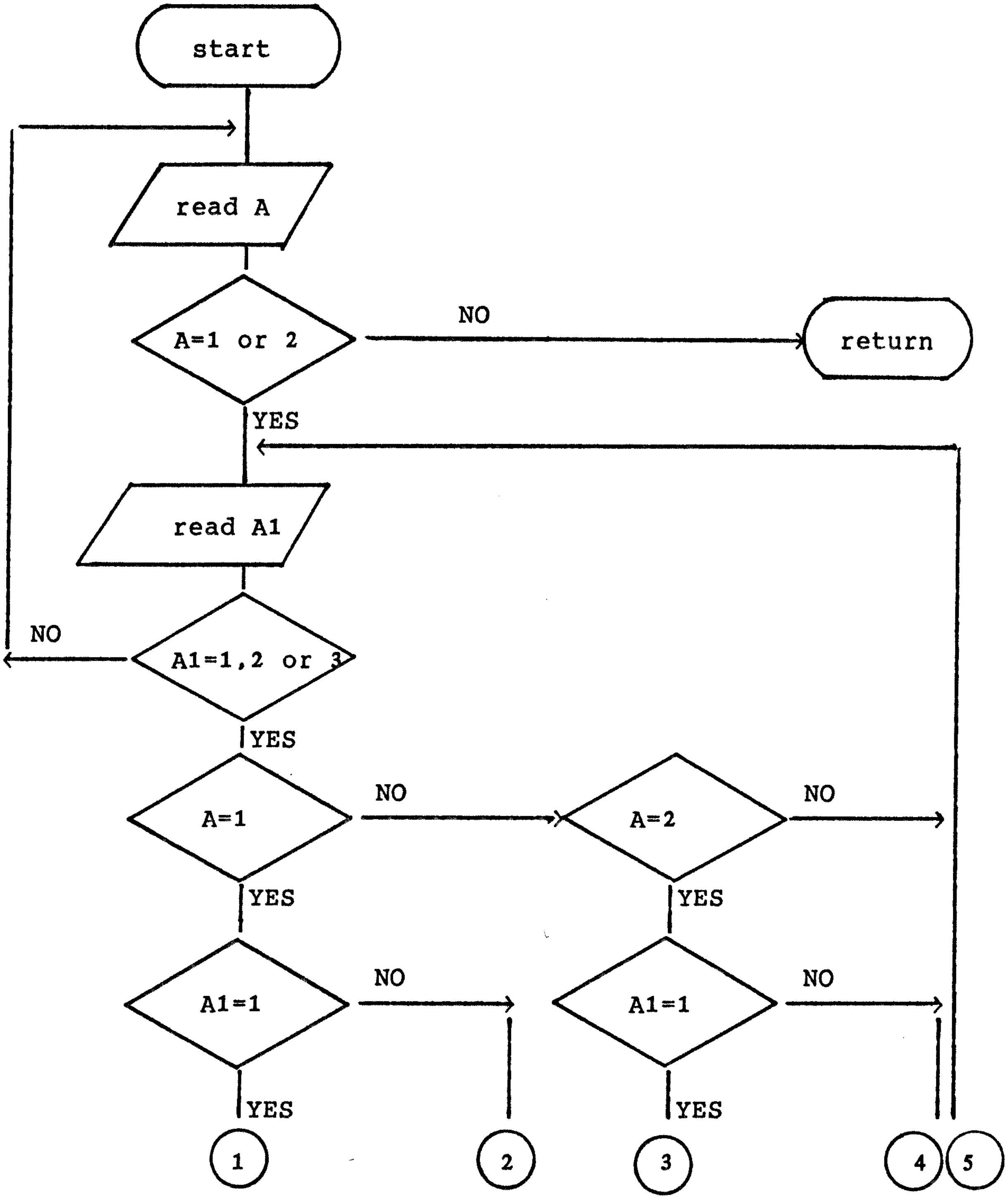

Figure II-B-12-a. Flow chart of the subroutine PARA.

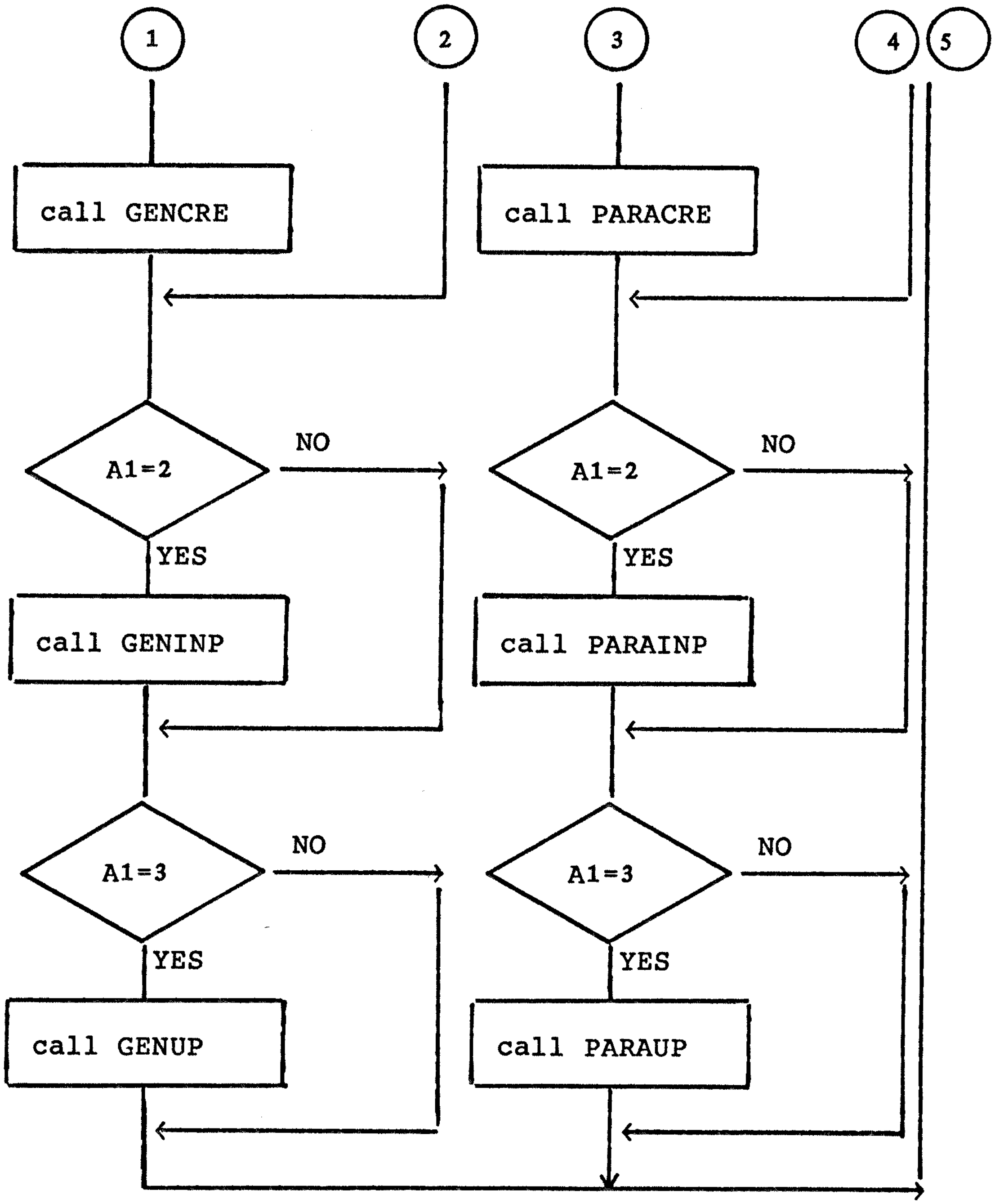

Figure II-B-12-b. Flow chart of the subroutine PARA.

19. GENCRE

Subroutine GENCRE creates a new file which contains the general properties of each compound. The user is warned to avoid using duplicate names in file creation, is asked for the file name to be created, and is asked to give the total number of compounds. After these steps, the file is created and opened. Control then returns to the calling program. Figure II-B-13 is the flow chart.

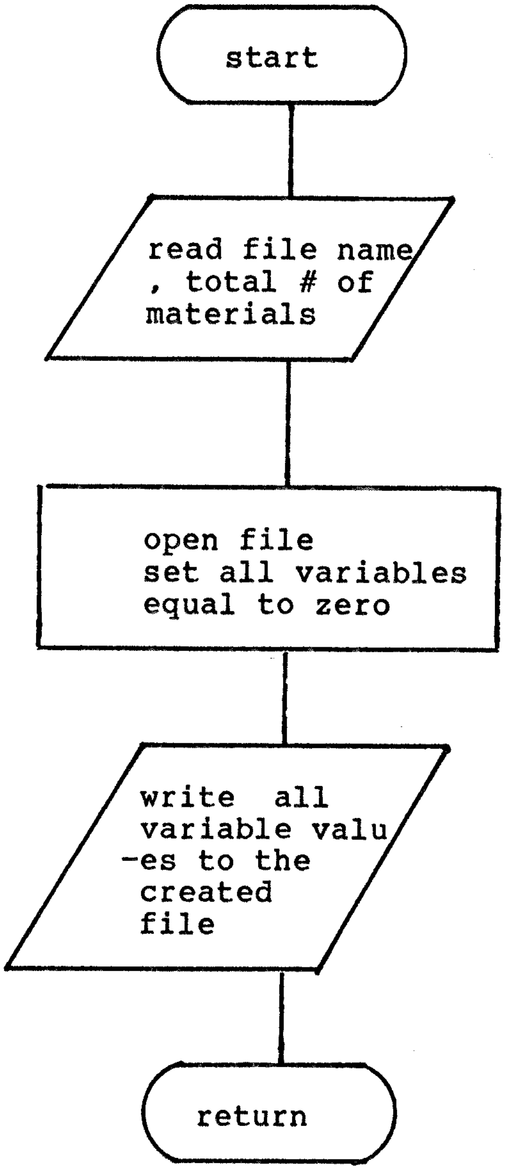

Figure II-B-13. Flow chart of the subroutine GENCRE.

20. GENINP

Subroutine GENINP is used to input data for the first time to the file created by subroutine GENCRE . The user is asked to enter the file name and the record number to be controlled. If the user gives 0, then control returns to the calling program. If the value is a positive integer, data are read interactively and recorded to the designated record number. The program then loops back to ask for the next record. The flow chart is given in Figure II-B-14.

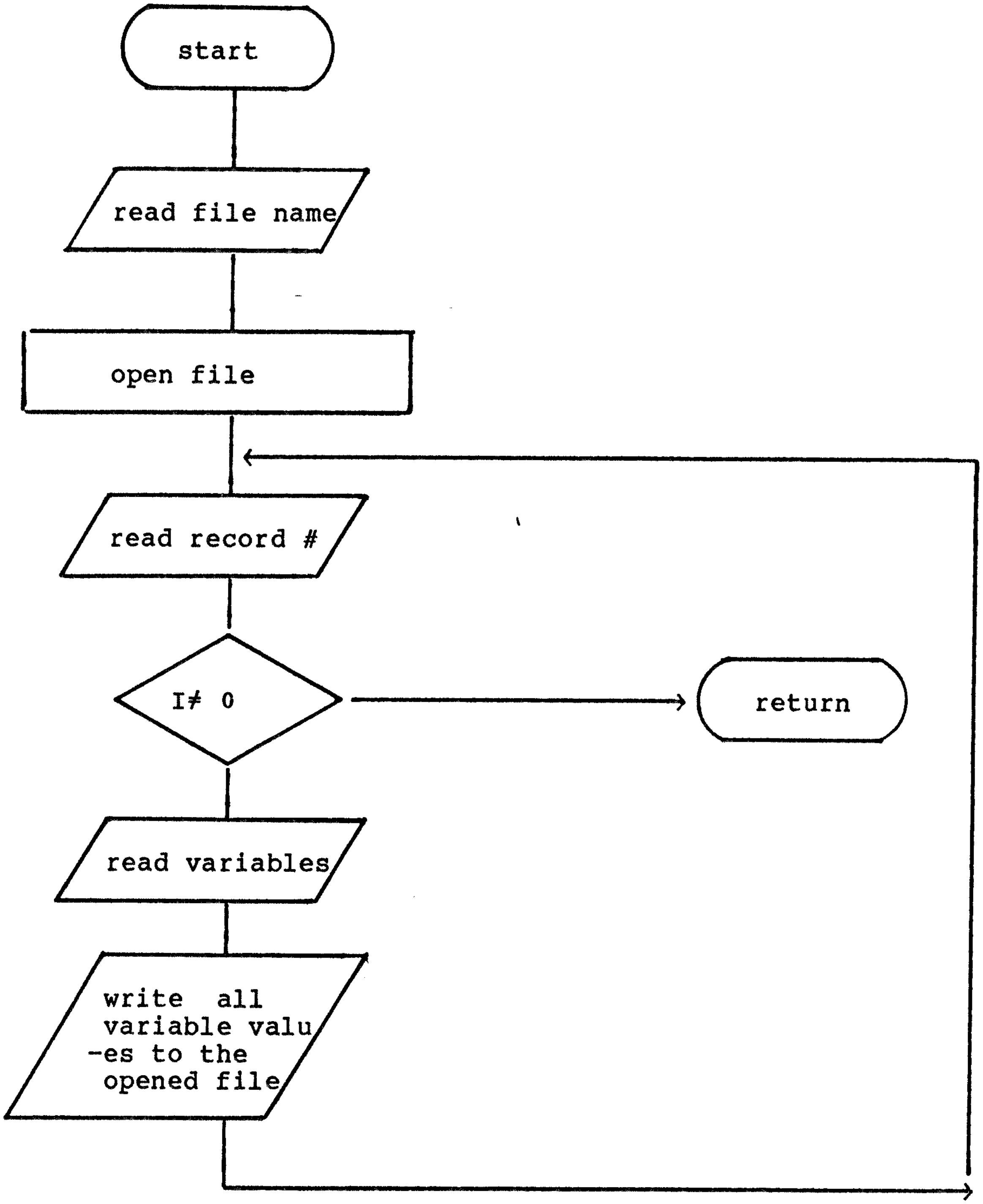

Figure II-B-14. Flow chart of the subroutine GENINP.

2-21. GENUP.

Subroutine GENUP is used to access and update files which were created by GENCRE and inputted by GENINP. The user is asked for the file name and the record number to be checked. If the user gives 0 or a negative integer, then control returns to the calling program. If the value is a positive integer, data are read from the file and the data values are shown on the screen. At this point, the user is asked if updating is necessary. If the user declines to update, the program asks for a new record number. If the user choose to update the data, it is done interactively. Upon completion, the user is asked for any additional record numbers. At this point, if user gives a negative response, then control returns to the calling program. If the user wishes to do further updating, then the procedure repeats itself. The flow chart is contained in Figure II-B-15.

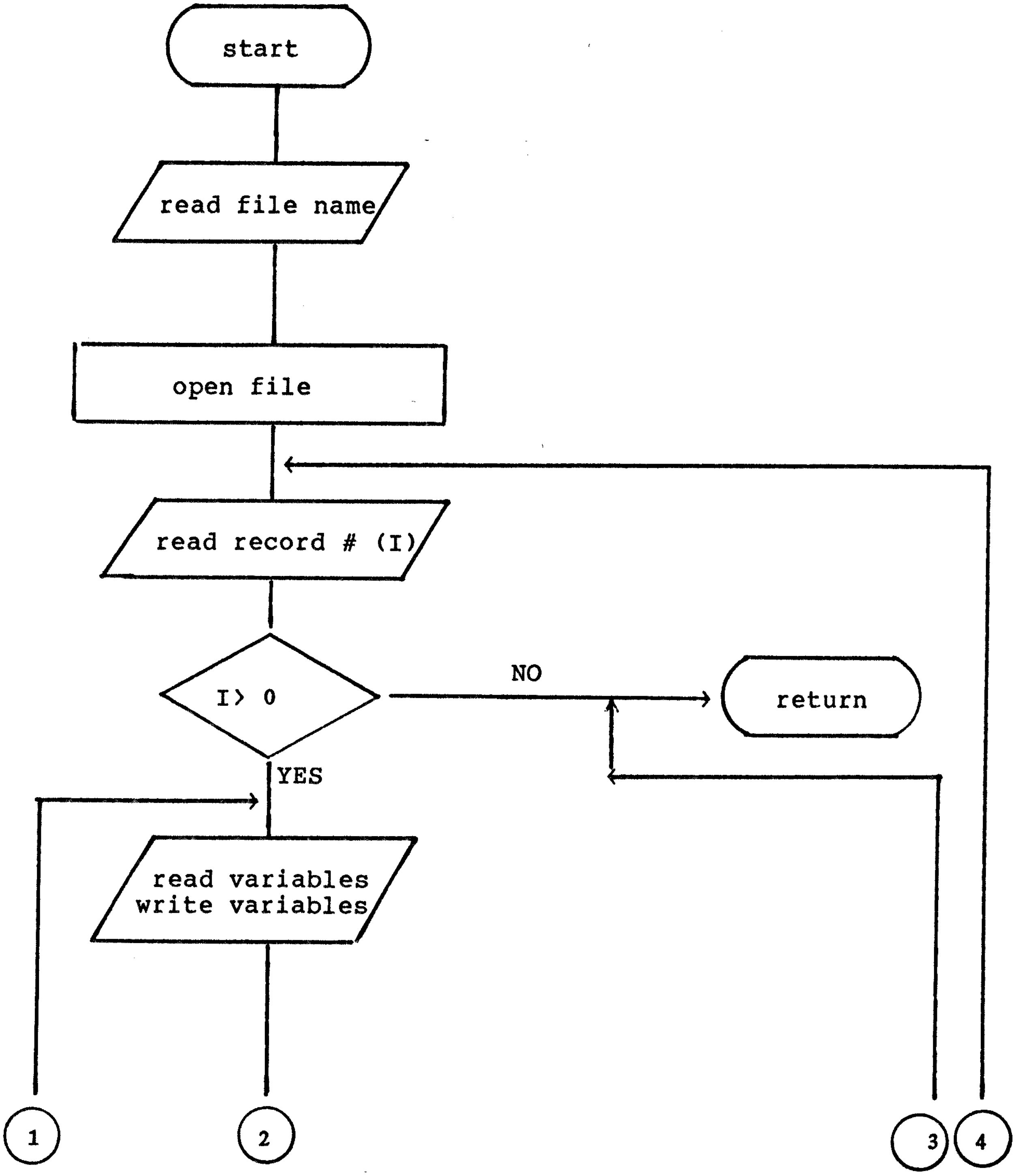

Figure II-B-15-a. Flow chart of the subroutine GENUP.

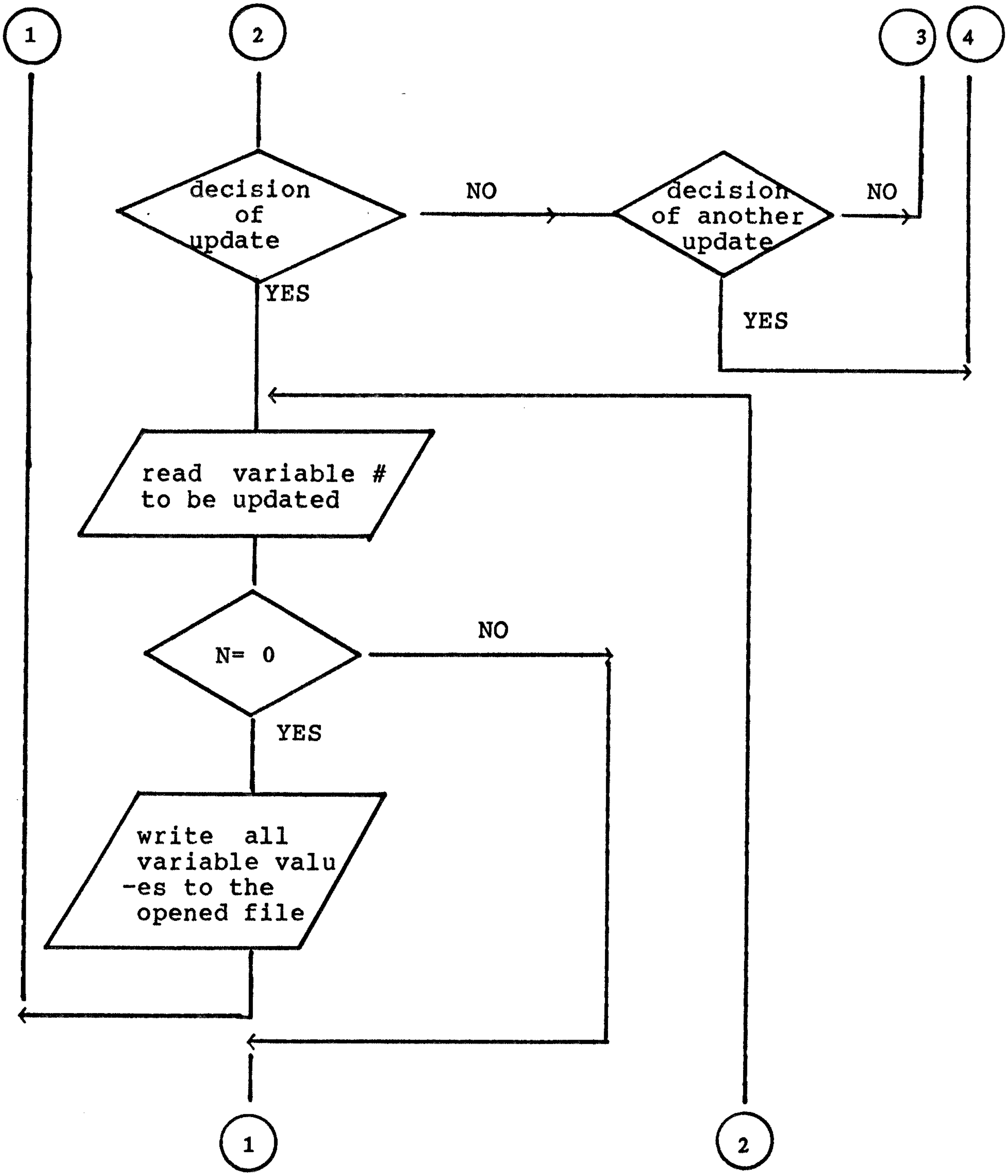
1
2
3
4
decision of update
NO
decision of another update
NO
YES
YES
read variable # to be updated
N= 0
NO
YES
write all variable valu -es to the opened file
1
2

Figure II-B-15-b. Flow chart of the subroutine GENUP.

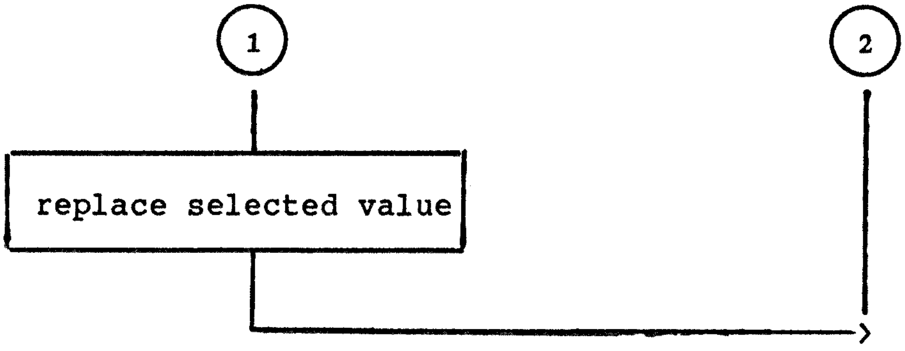

Figure II-B-15-c. Flow chart of the subroutine GENUP.

22. PARACRE

Subroutine PARACRE is the program for the purpose of creating the binary interaction parameter files. The user is warned to avoid using duplicate names in file creation, asked for the file name to be created, desired number of variables per record, total number of components and desired initial value for parameters. Then after this procedure, the file is created. Figure II-B-16 is the flow chart.

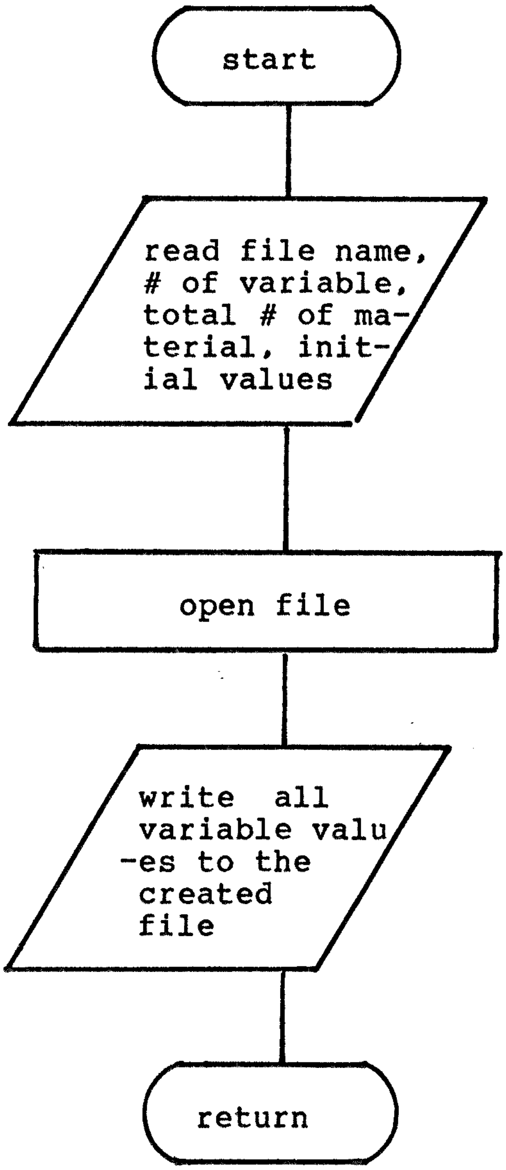
start
read file name, # of variable, total # of material, initial values
open file
write all variable valu -es to the created file
return

Figure II-B-16. Flow chart of subroutine PARACRE.

23. PARAINP

Subroutine PARAINP is used to input data to the file created by subroutine PARACRE. The user is asked for the file name, number of variables per record and opens the file. The user is also asked for the total # of systems, total # of components of each system and component ID. The user then enters the data, after which control returns to the calling program. Figure II-B-17 is the flow chart.

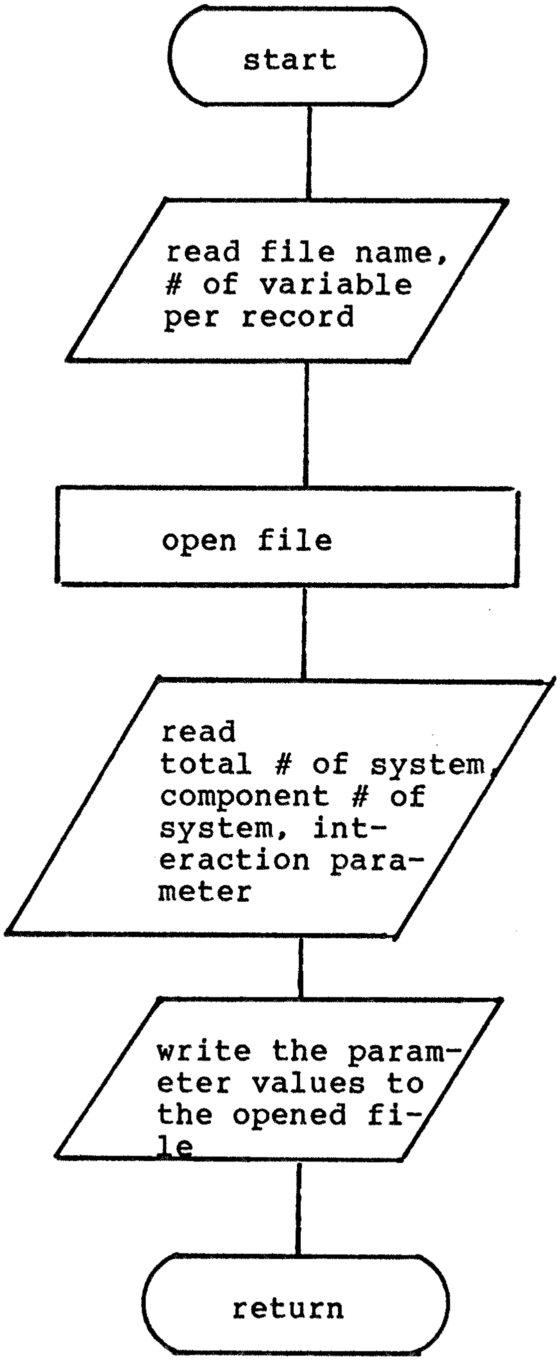

Figure II-B-17. Flow chart of the subroutine PARAINP.

24. PARAUP

Subroutine PARAUP is used to access and update the parameter files which were created by PARACRE and inputted by PARAINP. The user is asked for the file name and the number of variables per record. The procedure is quite similar to the subroutine GENUP. Figure II-B-18 is the flow chart of the program.

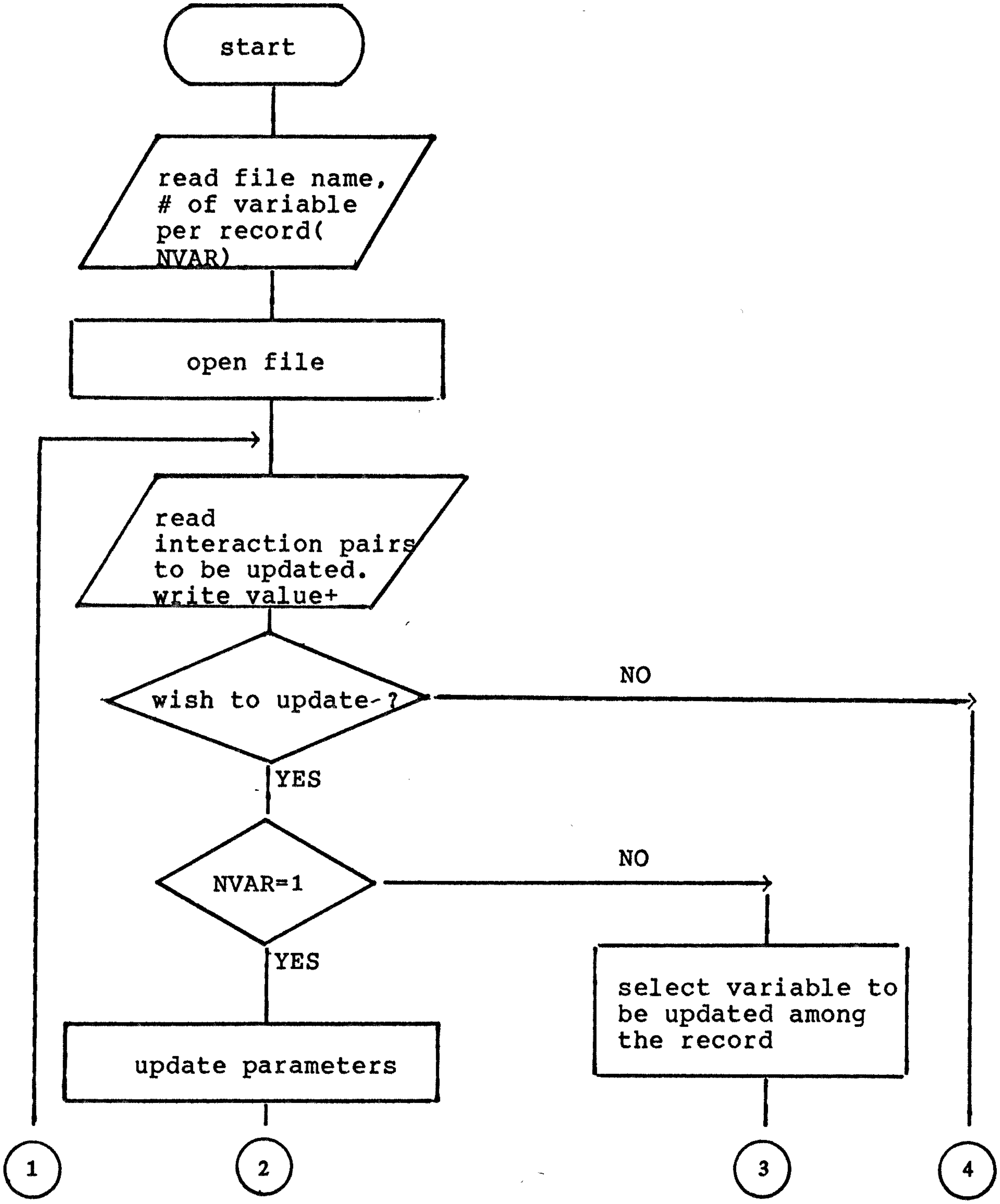

Figure II-B-18-a. Flow chart of the subroutine PARAUP.

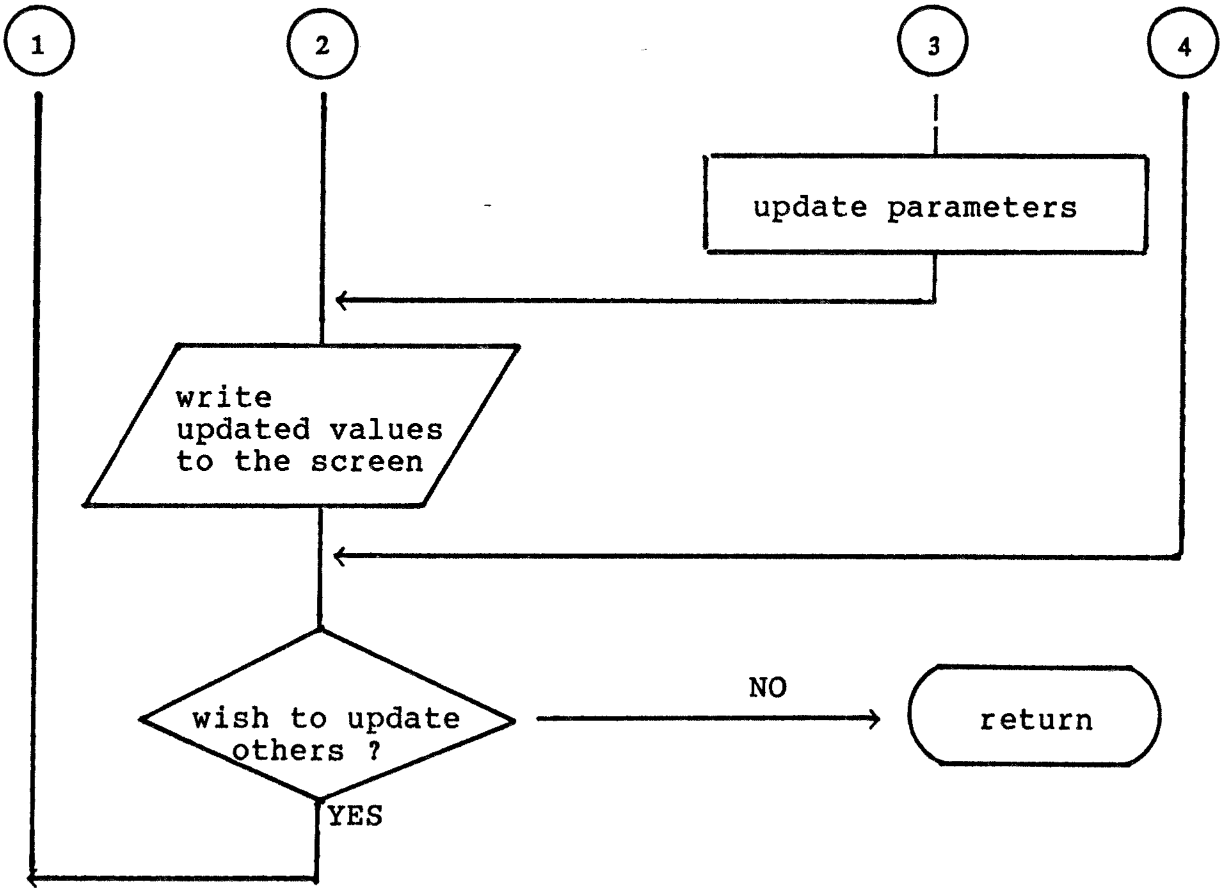

Figure II-B-18-b. Flow chart of the subroutine PARAUP.

II. Reliability test

A. Vapor/Liquid Equilibrium (Isothermal P-X) calculation.

This section gives sample results of dew point pressure calculations which represent one of the vapor/liquid equilibrium calculations. Temperature(T) and vapor composition(y) values are given and total pressure(P) and liquid composition(X) are calculated. Two test systems were chosen. One consists of the n-hexane, ethanol, methylcyclopentane and benzene quaternary system(11), for which the Wilson and modified UNIQUAC models were used for the calculation. The other one is the toluene and n-heptane binary system(12) which uses the NRTL model for the calculation. The results are given in tables A-1 and 2.

Table A-1

System:Hexane-Ethanol-Methylcyclopentane(MCP)-Benzene

Reference:(11)

(W: Wilson, mU: modified UNIQUAC)

Data Point	Pressure(atm)			n-hexane(X_1)			EtOH(X_2)			MCP(X_3)			C6H6(X_4)		
	exp	W	mU	exp	W	mU	exp	W	mU	exp	W	mU	exp	W	mU
(1) T=334.2 K y=0.169, 0.329, 0.279, 0.223	1.000	0.996	0.990	0.150	0.142	0.151	0.276	0.270	0.244	0.285	0.276	0.286	0.299	0.311	0.319
(2) T=334.0 K, y=0.145, 0.310, 0.417, 0.128	1.000	1.007	1.003	0.156	0.156	0.157	0.154	0.148	0.152	0.510	0.510	0.503	0.180	0.186	0.187
(3) T=338.5 K, y=0.717, 0.015, 0.058, 0.210	1.000	1.025	1.035	0.717	0.709	0.709	0.015	0.016	0.015	0.058	0.057	0.059	0.210	0.219	0.218
Error(%)	-	1.20	1.60	-	2.15	0.81	-	3.70	4.29	-	1.62	1.15	-	2.76	3.68

Table A-2

System: Toluene(1)- n-Heptane(2)

Model : NRTL

Reference:(12)

Data Point	Results			
(1) T=298.15 k	Pressure(atm)		Toluene (X_1)	
Y_1	exp	prediction	exp	prediction
0.0	0.06002	0.06034	0.0	0.0
0.059	0.06052	0.05999	0.066	0.066
0.138	0.05864	0.05928	0.161	0.159
0.216	0.05775	0.05829	0.261	0.258
0.303	0.05657	0.05680	0.377	0.375
0.442	0.05321	0.05356	0.564	0.565
0.490	0.05203	0.05223	0.623	0.627
0.605	0.04838	0.04876	0.751	0.756
0.734	0.04413	0.04476	0.860	0.866
0.929	0.03998	0.03920	0.971	0.973
1.000	0.03702	0.03740	1.000	1.000
(2) T=303.15 k				
0.0	0.07631	0.07703	0.0	0.0
0.059	0.07552	0.07661	0.066	0.065
0.138	0.07503	0.07576	0.161	0.158
0.216	0.07354	0.07581	0.261	0.257
0.304	0.07187	0.07268	0.377	0.374
0.445	0.06753	0.06855	0.564	0.566
0.493	0.06595	0.06687	0.623	0.627
0.609	0.06239	0.06247	0.751	0.757
0.738	0.05687	0.05742	0.860	0.867
0.931	0.05114	0.05044	0.971	0.973
1.000	0.04877	0.04821	1.000	1.000

(3)
T=313.15 k

0.0	0.09175	0.12200	0.0	0.0
0.059	0.12054	0.12140	0.066	0.064
0.138	0.11867	0.12020	0.161	0.157
0.217	0.11669	0.11840	0.261	0.255
0.306	0.11383	0.11560	0.377	0.373
0.449	0.10751	0.10920	0.564	0.565
0.498	0.10514	0.10660	0.623	0.627
0.640	0.09705	0.09821	0.773	0.782
0.746	0.09033	0.09178	0.860	0.869
0.934	0.08135	0.08115	0.971	0.974
1.000	0.07789	0.07778	1.000	1.000
Error(%)	-	1.95	-	0.79

B. Vapor/Liquid Equilibrium(Isothermal T-X) calculation.

This section shows results of the dew point temperature calculation routine. Total pressure(P) and vapor composition(y) values are given, and the program calculates temperature(T) and liquid composition(x_i) values. Three test systems were chosen. The first consists of nitrogen, argon and oxygen ternary system(13), with the Wilson model used for the calculation. Another one is acetone, methanol and chloroform ternary system(14), which uses the Wilson model. The third one is ethanol and water binary system(15). The Knox model is used for this system. The results are given in tables B-1,2 and 3.

Table B-1

System:Nitrogen-Argon-Oxygen

Reference:(13)

Model	P(Given)	T(Calculated)	T(Experimental)
Wilson	1.316 atm	88.85 K	89.08 K
Material	y(Given)	x(Calculated)	x(Experimental)
Nitrogen	0.3036	0.1106	0.1192
Argon	0.2948	0.3044	0.2992
Oxygen	0.4016	0.5850	0.5816
Error(%)	Composition:3.18	Temperature:0.26	

Table B-2

System:Acetone-Methanol-Chloroform

Model: Wilson

Reference:(14)

Data Point	Temperature (K)		Acetone(X_1)		MeOH(X_2)		Chloroform(X_3)	
	exp	prediction	exp	prediction	exp	prediction	exp	prediction
(1) P=0.6921 atm y=0.410, 0.243, 0.151	323.2	322.2	0.428	0.437	0.146	0.151	0.426	0.412
(2) P=0.6355 atm y=0.528, 0.066, 0.406	323.24	322.7	0.497	0.495	0.029	0.027	0.474	0.478
Error(%)	-	0.24	-	1.23	-	5.1	-	2.0

Table B-3

System:Ethanol-Water

Model: Knox

Reference:(15)

Data Point	Temperature (K)		EtOH(X_1)	
	exp	prediction	exp	prediction
(1) P=0.2641 atm y=0.700, 0.300	323.2	322.0	0.5660	0.5796
(2) P=0.2782 atm y=0.8064, 0.1936	323.2	322.4	0.7802	0.7747
(3) P=0.1820 atm y=0.4127, 0.5873	323.2	321.7	0.0743	0.0690
Error (%)	-	1.08	-	3.40

C.Heat-of-Mixing calculation.

This section gives results for the heat of mixing(excess enthalpy) in liquid mixtures. Temperature(T) and liquid composition(x_i) values are given and the heat of mixing is calculated. Two binary test systems were chosen. The first consists of benzene and acetonitrile, and the Redlich-Kister equation is used for the calculation. The second is the benzene/cyclohexane system, and uses the Knox model for the calculation. The results are given in table C-1 and 2.

Table C-1

System:Benzene-Acetonitrile

Reference:(16)

Model: Redlich-Kister T(Given): 298.15 K

Composition(Given)		H^E(cal/mol)	
Benzene	Acetonitrile	Calculated	Experimental
0.9248	0.0752	25.0	25.7
0.8965	0.1035	33.5	33.4
0.8616	0.1384	43.4	43.0
0.8159	0.1841	55.3	54.2
0.7607	0.2393	68.3	69.8
0.6944	0.3056	81.8	81.4
0.6187	0.3813	94.5	94.3
0.5367	0.4633	104.4	104.2
0.4537	0.5463	109.8	110.0
0.3713	0.6287	109.5	109.1
0.2958	0.7042	103.4	103.9
0.2315	0.7685	92.9	93.1
0.1770	0.8230	79.5	79.2
0.1336	0.8664	65.4	65.4
0.0730	0.9270	40.2	40.0
Error(%)		0.74	

Table C-2

System:Benzene-Cyclohexane

Reference:(17)

Model: Knox T(Given): 298.15 K

Composition(Given)		H^E(cal/mol)	
Benzene	Cyclohexane	Calculated	Experimental
0.02055	0.97945	63.48	62.71
0.10063	0.89937	292.10	283.30
0.22787	0.77213	585.20	552.61
0.37522	0.62478	800.70	741.81
0.49163	0.50837	866.60	798.72
0.63650	0.36350	812.10	751.99
0.78027	0.21973	605.20	571.43
0.89570	0.10430	329.80	319.37
0.98720	0.01280	44.50	44.10
Error(%)		4.97	

D. Liquid/Liquid Equilibrium separation calculation.

This section shows the results of liquid/liquid equilibrium separation calculations. Temperature(T) and feed composition values are given and the two liquid phase compositions at equilibrium are calculated. One phase is called extract phase, the other phase is called raffinate phase. Four systems were chosen. The first consists of furfural, 2,2,4-trimethyl pentane, benzene and cyclohexane quaternary system, and the UNIQUAC model is used for the calculation. The second is water, acrylonitrile and acetonitrile ternary system, using the modified UNIQUAC model. The third is furfural, 2,2,4- trimethyl pentane and cyclohexane ternary system. The fourth is furfural, 2,2,4-trimethyl pentane and benzene. The UNIQUAC model is used for both the third and fourth systems. The difference between this section and the others is that there are no experimental data. But there is a reference(18) in which the calculations for the same systems are done by others. The results are given in tables D-1,2,3,4.

Table D-1

System:Furfural-2,2,4 trimethyl pentane-Benzene-
Cyclclohexane(Quaternary system)

Model: UNIQUAC

Reference:(18)

T(Given) 298.15 K

R: raffinate phase
E: extract phase

Material	Feed mole fraction	calculated		reference	
		R	E	R	E
Furfural	0.3	0.605	0.160	0.604	0.160
2,2,4 trimethyl pentane	0.1	0.031	0.131	0.032	0.131
Benzene	0.2	0.187	0.206	0.187	0.206
Cyclohexane	0.4	0.177	0.502	0.177	0.502
Furfural	0.4	0.788	0.091	0.788	0.091
2,2,4 trimethyl pentane	0.2	0.033	0.333	0.033	0.333
Benzene	0.1	0.090	0.108	0.090	0.108
Cyclohexane	0.3	0.089	0.468	0.089	0.468
Furfural	0.4	0.814	0.086	0.814	0.086
2,2,4 trimethyl pentane	0.3	0.044	0.495	0.044	0.494
Benzene	0.1	0.088	0.109	0.088	0.109
Cyclohexane	0.2	0.054	0.310	0.054	0.311

Table D-2

System:Water-Acrylonitrile-Acetonitrile

Model: modified UNIQUAC

Reference:(18)

	T(Given) 333.15 K	R: raffinate phase E: extract phase			
Material	Feed mole fraction	calculated		reference	
		R	E	R	E
Water	0.60	0.952	0.214	0.952	0.214
Acrylonitrile	0.35	0.034	0.696	0.034	0.696
Acetonitrile	0.05	0.014	0.090	0.014	0.090
Water	0.60	0.837	0.469	0.837	0.469
Acrylonitrile	0.15	0.043	0.209	0.043	0.209
Acetonitrile	0.25	0.120	0.322	0.120	0.322

Table D-3

System:Furfural-2,2,4 trimethyl pentane-Cyclohexane

Model:UNIQUAC

Reference:(18)

	T(Given) 298.15 K	R: raffinate phase E: extract phase			
Material	Feed mole fraction	calculated		reference	
		R	E	R	E
Furfural	0.4	0.914	0.057	0.914	0.058
2,2,4 tri-methyl pentane	0.3	0.028	0.482	0.028	0.481
Cyclohexane	0.3	0.058	0.461	0.058	0.461
Furfural	0.1	0.873	0.059	0.872	0.059
2,2,4 tri-methyl pentane	0.1	0.007	0.105	0.008	0.105
Cyclohexane	0.8	0.120	0.837	0.120	0.836

Table D-4

System:Furfural-2,2,4 trimethyl pentane-Benzene

Model: UNIQUAC

Reference:(18)

	T(Given) 298.15 K	R: raffinate phase E: extract phase			
Material	Feed mole fraction	calculated		reference	
		R	E	R	E
Furfural	0.4	0.740	0.112	0.740	0.112
2,2,4 trimethyl pentane	0.4	0.080	0.672	0.080	0.672
Benzene	0.2	0.181	0.216	0.181	0.216
Furfural	0.400	0.537	0.196	0.537	0.197
2,2,4 trimethyl pentane	0.270	0.144	0.458	0.144	0.457
Benzene	0.330	0.319	0.346	0.319	0.346

III. Discussion

There are so many kinds of experimental data which can be predicted by this program but few among them were chosen. The chosen data represent or imply the characteristics of each calculation or model. At the same time there are also many experimental data which can not be predicted due to the lack of parameter values. As the regression function of this program is not yet completed, it can not yet generate the parameter values from experimental data. Parameter values must be supplied from other references. As a result, not every experiment can simulated. The following are detailed discussion of each case which was listed.

A. Vapor/Liquid Equilibrium(Isothermal P-X) calculation

This section discusses the dewpoint pressure calculation, a relatively easy type of prediction.

Table A-1 refers to a quaternary system containing alcohol. Both the Wilson and the modified UNIQUAC model are used, and the results are compared with each other. These four materials are all condensable at the temperatures 334.2 K, 334.0 K and 338.5 K. The test results at this temperatures show less than 5% error. Wilson model shows good fit for about one hundred completely miscible sytems which include hydrocarbons, alcohols, ethers, esters, ketones, water and nitrogen-, sulfur-, and halogencontaining compounds. This is the main reason why the Wilson model works well for this system. But it has the limitation that the model can't be used for partially miscible, i.e., liquid/liquid equilibrium, systems.

The UNIQUAC and modified UNIQUAC models can be used without such a limitation. At first, when the UNIQUAC model was used in this system, the results were very poor. As indicated previously, the UNIQUAC model shows very poor results for systems including either alcohol or water. In this case, the modified UNIQUAC model(which contains one more parameter value for these components) shows better accuracy. Thus, the modified UNIQUAC model is used for this system. But basically there are no differences between these two models.

Another vapor/liquid equilibrium calculation involving noncondensable components was performed with the Wilson model. The system was composed of Methane, n-Butane, Decane, but the results did not agree well with the experimental data. Two possible reasons exist. One possibility is incorrect programming, and another is unreliable experimental data. The test results for this case were not included in the previous section for this reason. Both possible explanations must be investigated in further research.

Table A-2 shows the results of toluene and n-heptane binary system calculation by the NRTL model at temperatures

298.15 K, 303.15 K and 313.15 K. The results indicate an excellent fit to the data over the entire temperature range. The reason for this is that this system deals only with binary components, while the former system is quaternary. The NRTL model can also be applied to liquid/liquid systems but it uses an additional parameter value named α. In theory this α value must be positive, but sometimes a negative α value(19) gives better results, especially α=-1.0. This problem is associated with the formulation of the NRTL model.

Until now several successful results are obtained using subroutine DEWP, but also a few unsuccessful cases were encountered for systems containing noncondensable components. As indicated before, another algorithm(20) may also be tried in further research.

B. Vapor/Liquid Equilibrium(Isothermal T-X) calculation

This section discusses the dewpoint temperature calculation. This kind of prediction is more difficult than the dewpoint pressure calculation. The main problem is how to guess the temperature, and how to adjust it during the iterative procedure. The present algorithm for temperature convergence needs many trials to get the correct temperature. This problem is serious in the view point of numerical analysis. In future versions of this program, this procedure must be modified. Fixed- point iteration and the Newton-Raphson method or its modifications have some possibilities for the correction of the program. Also, the algorithm introduced in section A should be considered to reduce the program size.

Table B-1 shows the calculated results of Nitrogen, Argon and Oxygen ternary system by using Wilson model at pressure 1.316 atm. The temperature calculation error is much smaller than the composition error. This means that the present algorithm shows the good accuracy, and at the same time the Wilson model shows a good prediction ability for the nitrogen-argon-oxygen containing system, which has a very low boiling point. But one thing to heed is that the system temperature is still below the critical point. For temperature higher than the critical point of one or more of the components, it is difficult to get a reliabile prediction results.

Table B-2 shows calculated results of ternary system consists of Acetone, Methanol and Chloroform by using Wilson model at the pressure of 0.6921, 0.6355 atm. This table also shows good accuracy. But this system implies the limitation of the Wilson model. Wilson parameters are required for this system but it is not possible to find values of the Wilson parameters which reproduce maxima and minima in the activity coefficient. This behavior is encountered in mixtures of chloroform with alcohols and this system is such a case. Fortunately this is not a serious defect since the equations do provide a good approximation for the activity coefficient over essentially all of the composition range, as shown for this system. Sets of Wilson parameters have been found which lead to maxima and minima in ln(γ) where γ is a activity coefficient, but such behavior appears to be confined to virtually ideal system, i.e., to a very narrow band of γ values between 0.8 and 1.2. In this region other sets of parameters exist which yield a monotonic dependence of γ on liquid composition x.

Table B-3 shows calculated results of binary systems consist of ethanol and water, using the Knox model at pressures of 0.2641, 0.2782 and 0.1820 atm. This table also shows good accuracy. As introduced before the correct name of Knox model is Knox Molecular model. But in this thesis, this is defined as the Knox model for the convenience. This model is based on different concepts from Wilson, NRTL and UNIQUAC model. The Knox model can be applied to all 4 kinds of calculation in this thesis and shows good results. But until now, few parameter values are available while the others can supply abundant parameter values. Parameter evaluation for this model needs to be done.

C. Heat of Mixing (Liquid mixture) calculation

This section discusses the heat of mixing(h^E) calculation. This kind of prediction is the most difficult type of calculation. The results calculated with given parameter rarely shows consistency with experimental data. At first Wilson model gives very poor results. And for the case of UNIQUAC and NRTL, the suitable parameter values are not found. The models that can be used for this purpose are Redlich-Kister and Knox equations. Until now, the other models are not satisfactory. The Redlich-Kister model shows the most the precise results for this purpose. But it needs so many parameter values(generally more than 3) and these parameter values are sensitive to temperature change that a new parameter set must be supplied at each temperature. This is a serious disadvantage of this model and this model is not suitable for the other 3 kinds of calculation for this reason. The Knox model can be used with the same parameters within some temperature ranges.

Table C-1 shows calculated results of binary system consisting of Benzene and Acetonitrile using Redlich-Kister model at a temperature of 298.15 K. The error is very small but this calculation is possible only at the temperature associated with the parameter values. To solve this problem another type of model like Wilson, NRTL, UNIQUAC, Knox model must be used. Among these, the Knox model shows the better results for this purpose.

Table C-2 shows also binary systems of benzene and cyclohexane using the Knox model. It gives good results but a little less accuracy than Redlich-Kister. More parameter values must be developed for wider ranges and various mixtures.

D. Liquid/Liquid equilibrium separation calculation

This section contains the liquid/liquid equilbrium calculation results. This kind of prediction is more difficult than vapor/liquid equilibrium calculations. Here, the UNIQUAC and the modified UNIQUAC models are used. The other models do not have the necessary parameters available, and also few experimental data are available with the result that only limited models can be used. Comparison is made with the results of another reference(18), which treated the same case.

The results are essentially the same as in the reference from Table D-1 to D-4. Table D-2 is for the modified UNIQUAC model because the system contains water which needs specific parameter value.

Generally for liquid/liquid equilibrium calculations the rate of solution convergence is strongly affected by the distance of the feed composition from any plait point region. As the feed composition approaches a plait point, the rate of convergence of the calculation procedure is markedly reduced. Typically, 10 to 20 iterations are required, this happens in the last case of Table D-2 and also last case of Table D-4. Very near a plait point, convergence can be extremely slow, requiring 50 iterations or more. In this case the program terminates without a solution, but this is not a significant disadvantage since liquid/liquid separations are not intentionally conducted near plait points.

IV. Conclusions and suggestions

1) In vapor-liquid equilibrium calculation, the reliability of this program is very good. However the algorithm to perform the calculation of condensable and noncondensable components must be improved. Also, new algorithms to reduce the execution time, and to perform the other uncompleted vapor-liquid equilibrium calculations, should be finished.

2) In liquid-liquid equilibrium separations, good results are obtained, but some more sytems should be checked by using equations other than UNIQUAC to confirm the reliability of this part of the program.

3) Heat of mixing calculations shows relatively poor results except for the Knox and Redlich-Kister equations. This implies that presently developed modelds which are adopted in this program are generally poor for this purpose. Though Redlich-Kister equation shows good fitting ability, it has some severe limitations for practical use.

4) Among the equations for the calculation of activity coefficients for multicomponent systems, the Knox equation has the most firm theoretical basis and the UNIQUAC equation

is powerful in practical use for versatile systems, but all the systems can not be treated by one kind of model and each calculation (vapor-liquid, liquid-liquid and heat of mixing) requires different parameter values even though it is only one model. This is a vexatious thing in data input. Next, the subroutines for calculate of activity coefficients from equations of state must be finished.

5) The parameter regression subroutine should be developed.

6) In future, models which can treat electrolyte system and models to perform all calculations using the same parameter values at the same condition are to be developed and introduced to this program.

V. References.

1. Hayden, J.G., O'Connell, J.P., Ind. Eng. Chem., Process Des. Dev., 14, 221 (1975).
2. Nothnagel, K.H., Abrams, D.S., Prausnitz, J.M., Ind. Eng. Chem., Process Des. Dev., 12, 25 (1973).
3. Spencer, C.F., Danner, R.P., J. Chem. Eng. Data, 17, 236 (1972).
4. Lyckman, E., Eckert, C.A. and Prausnitz, J.M., Chem. Eng. Sci., 20, 685, 703 (1965).
5. Knox, D.E., Hollinger, H.B. and Van Ness, H.C., Fluid Phase Equilibria, 15, 267 (1984).
6. Abrams, D.S., Prausnitz, J.M., A.I.Ch.E. J., 21, 116 (1975).
7. Anderson, T.F., Abrams, D.S., Grens, E.A., A.I.Ch.E. J., 24, 20 (1978).
8. Renon, H. and Prausnitz, J.M., A.I.Ch.E. J., 14, 135 (1968).
9. Wilson, G.M., J. Am. Chem. Soc., 86, 127 (1964).
10. Redlich, O., Kister, A.T. and Turnquist, C.E., Chem. Eng. Progr. Symp. Ser., 48, 49 (1952).
11. Prausnitz, J.M., Eckert, C.A., Orye, R.V., O'Connell, J.P., Computer Calculations for Multicomponent Vapor-Liquid Equilibria (N.J.: Prentice-Hall, 1967).
12. Stanley, J.A., Andrew, D.C. and Ronald, B.S., J. Am. Chem. Soc, 24, 195 (1979).

13. Latimer, R.E., Amer. Inst. Chem. Eng. J., 3, 75 (1957).

14. Severns, Jr., W.H., et al., Amer. Inst. Chem. Eng. J., 1, 401 (1955).

15. Wilson, S.R., Patel, R.B., Abbott, M.M. and Van Ness, H.C., J. Chem. Eng. Data, 24, 130 (1979).

16. Afaf, H.A., et al, J. Chem. Eng. Data, 21, 304 (1976).

17. Marsh, K.N., Int. Data Ser., Sel. Data. Mixtures, A, 3, 23 (1973).

18. Prausnitz, J.M., Anerson, T., Grens, E., Eckert, C.A., Hi Hsieh, R., O'Connell, J.P., Computer Calculations for Multicomponent Vapor-Liquid Liquid-Liquid Equilibria (N.J.: Prentice-Hall, 1980).

19. Marina, J.M. and Tassios, D.P., Ind. Eng. Chem. Process. Des. Dev., 12, 67 (1973).

20. Stanley, M.W., Phase Equilibria in Chemical Engineering (Boston: Butterworth Publishers, 1985).

Appendix

PROGRAM LIST

```
      PROGRAM MASTER
C
C       This is the master control Program for the regression and
C         Prediction of thermodynamic Properties of mixtures using
c         ANY of a variety of thermodynamic models.  The program
C         may also be used to update current values of model parameters.
C
      INTEGER GOAL
      CHARACTER*1 ANS,BEEP,A
      CHARACTER*3 ANS3,NBR3(500),NBR2(99),NBR1(9)
      CHARACTER*3 NBRA(100),NBRB(100),NBRC(100),NBRD(100),NBRE(100)
      CHARACTER*100 NBRS
      CHARACTER*100 NBRT
      COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX
     +,MIXCOM(50,10),NCM(10)
      COMMON/CNTRL/MODEL,GOAL,NPROP,NC,NG
       EQUIVALENCE (NBR3(1),NBRA(1)),(NBR3(101),NBRB(1)),
     + (NBR3(201),NBRC(1)),(NBR3(301),NBRD(1)),(NBR3(401),NBRE(1))
C      Maximum # mixtures = 50, max # components = 20, max comp per mix
       DATA MAXMIX/50/,MAXCOM/20/,MXCMMX/10/,NCOMPS/468/
      NBRS='1234567890AB
     +                                '

      NBRT='101112131415161718192021222324252627282930313233343536373839
     +4041424344454647484950'
      BEEP=CHAR(7)
  201 GOAL=0
        WRITE(*,1)
        READ(*,11,END=301) ANS
        DO 101 I=1,4
          IF(ANS.EQ.NBRS(I:I)) GOAL=I
  101   CONTINUE

        IF(GOAL.EQ.4) STOP
        IF(GOAL.EQ.3) THEN
        CALL SORT
        GO TO 201
        END IF
        IF(GOAL.NE.0) GO TO 202
  301     WRITE(*,1001) BEEP,BEEP
          GO TO 201
  202 NPROP=0
        WRITE(*,2)
        READ(*,11,END=302) ANS
        DO 102 I=1,5
          IF(ANS.EQ.NBRS(I:I)) NPROP=I
  102   CONTINUE
```

```
       IF (NPROP.EQ.5) GO TO 201
       IF (NPROP.NE.0) GO TO 203
 302     WRITE(*,1001) BEEP,BEEP
         GO TO 202
 203 MODEL=0
       WRITE(*,3)
       READ(*,11,END=303) ANS
       DO 103 I=1,12
         IF(ANS.EQ.NBRS(I:I)) MODEL=I
 103   CONTINUE
       IF (MODEL.EQ.12) GO TO 202
       IF (MODEL.EQ.11) GO TO 204
       IF (MODEL.NE.0) GO TO 205
 303     WRITE(*,1001) BEEP,BEEP
         GO TO 203
 204 MODEL=0
       WRITE(*,4)
       READ(*,11,END=304) ANS
       DO 104 I=1,12
         IF(ANS.EQ.NBRS(I:I)) MODEL=10+I
 104   CONTINUE
       IF (MODEL.EQ.22) GO TO 202
       IF (MODEL.EQ.21) GO TO 203
       IF (MODEL.NE.0) GO TO 205
 304     WRITE(*,1001) BEEP,BEEP
         GO TO 204
 205 NMIX=0
       WRITE(*,5) MAXMIX
       READ(*,12,END=305) ANS3
       IF(ANS3.EQ.'0  ') GO TO 305
       DO 105 I=1,MAXMIX
         IF(ANS3.EQ.NBRS(I:I)) NMIX=I
         IF(ANS3.EQ.NBRT(2*I-19:2*I-18)) NMIX=I

 105   CONTINUE

       GO TO 206
 305      WRITE(*,1001) BEEP,BEEP
          GO TO 204
 206 DO 2061 I=1,MAXMIX
       DO 2061 J=1,MXCMMX
         MIXCOM(I,J)=0
2061     CONTINUE
     DO 106 IMIX=1,NMIX
 207   NCOMP=0
       WRITE(*,6)IMIX,IMIX,MXCMMX,MAXCOM
       READ(*,12,END=306) ANS3
       IF(ANS3.EQ.'0  ') GO TO 306
       DO 107 I=1,MXCMMX
         IF(ANS3.EQ.NBRS(I:I)) NCOMP=I
       IF(ANS3.EQ.NBRT(2*I-19:2*I-18)) NCOMP=I

 107   CONTINUE

       IF(NCOMP.EQ.0) GO TO 306
       GO TO 208
 306  WRITE(*,1001) BEEP,BEEP
      GO TO 205
 208  DO 108 I=1,NCOMP

 501   ICOMP=0
       WRITE(*,701) IMIX
       WRITE(*,702)
       WRITE(*,703) I
      READ(*,*,ERR=601,END=601) ICOMP
        IF(ICOMP.LT.-12 .OR. ICOMP.GT.NCOMPS) GO TO 601
        IF(ICOMP.GT.0) GO TO 209
        IF(ICOMP.EQ.0) GO TO 505
        IGO=-1*ICOMP
        GO TO (501,505,509,513,517,521,525,529,533,537,541,545),IGO

 601   WRITE(*,*) BEEP,BEEP
       GO TO 501
 505   ICOMP=0
       WRITE(*,705) IMIX
       WRITE(*,706)
       WRITE(*,707) I
       READ(*,*,ERR=605,END=605) ICOMP
        IF(ICOMP.LT.-12 .OR. ICOMP.GT.NCOMPS) GO TO 605
        IF(ICOMP.GT.0) GO TO 209
        IF(ICOMP.EQ.0) GO TO 509
        IGO=-1*ICOMP
        GO TO (501,505,509,513,517,521,525,529,533,537,541,545),IGO
 605   WRITE(*,*) BEEP,BEEP
       GO TO 505
 509   ICOMP=0
       WRITE(*,709)IMIX
       WRITE(*,710)
       WRITE(*,711)I

       READ(*,*,ERR=609,END=609) ICOMP
        IF(ICOMP.LT.-12 .OR. ICOMP.GT.NCOMPS) GO TO 609
        IF(ICOMP.GT.0) GO TO 209
        IF(ICOMP.EQ.0) GO TO 513
        IGO=-1*ICOMP
        GO TO (501,505,509,513,517,521,525,529,533,537,541,545),IGO
 609   WRITE(*,*) BEEP,BEEP
       GO TO 509
 513   ICOMP=0
       WRITE(*,713) IMIX
       WRITE(*,714)
       WRITE(*,715) I
       READ(*,*,ERR=613,END=613) ICOMP
         IF(ICOMP.LT.-12 .OR. ICOMP.GT.NCOMPS) GO TO 613
         IF(ICOMP.GT.0) GO TO 209
         IF(ICOMP.EQ.0) GO TO 517
         IGO=-1*ICOMP
     GO TO (501,505,509,513,517,521,525,529,533,537,541,545),IGO
613    WRITE(*,*) BEEP,BEEP
       GO TO 513
517    ICOMP=0
       WRITE(*,717) IMIX
       WRITE(*,718)
       WRITE(*,719) I
       READ(*,*,ERR=617,END=617) ICOMP
         IF(ICOMP.LT.-12 .OR. ICOMP.GT.NCOMPS) GO TO 617
         IF(ICOMP.GT.0) GO TO 209
         IF(ICOMP.EQ.0) GO TO 521
         IGO=-1*ICOMP
         GO TO (501,505,509,513,517,521,525,529,533,537,541,545),IGO
617    WRITE(*,*) BEEP,BEEP
       GO TO 517
521    ICOMP=0
       WRITE(*,721) IMIX
       WRITE(*,722)
111    WRITE(*,723) I
       READ(*,*,ERR=621,END=621) ICOMP
         IF(ICOMP.LT.-12 .OR. ICOMP.GT.NCOMPS) GO TO 621
         IF(ICOMP.GT.0) GO TO 209
         IF(ICOMP.EQ.0) GO TO 525
         IGO=-1*ICOMP
         GO TO (501,505,509,513,517,521,525,529,533,537,541,545),IGO
621    WRITE(*,*) BEEP,BEEP
       GO TO 521


525    ICOMO=0
       WRITE(*,725)IMIX
       WRITE(*,726)
       WRITE(*,727)I
       READ(*,*,ERR=625,END=625) ICOMP
         IF(ICOMP.LT.-12 .OR. ICOMP.GT.NCOMPS) GO TO 625
         IF(ICOMP.GT.0) GO TO 209
         IF(ICOMP.EQ.0) GO TO 529
         IGO=-1*ICOMP
         GO TO (501,505,509,513,517,521,525,529,533,537,541,545),IGO
625    WRITE(*,*) BEEP,BEEP
       GO TO 525

529    ICOMP=0
```

```
      WRITE(*,729) IMIX
      WRITE(*,730)
      WRITE(*,731) I
      READ(*,*,ERR=629,END=629) ICOMP
        IF(ICOMP.LT.-12 .OR. ICOMP.GT.NCOMPS) GO TO 629
        IF(ICOMP.GT.0) GO TO 209
        IF(ICOMP.EQ.0) GO TO 533
        IGO=-1*ICOMP
        GO TO (501,505,509,513,517,521,525,529,533,537,541,545),IGO
629   WRITE(*,*) BEEP,BEEP
      GO TO 529
533   ICOMP=0
      WRITE(*,733) IMIX
      WRITE(*,734)
      WRITE(*,735) I
      READ(*,*,ERR=633,END=633) ICOMP
        IF(ICOMP.LT.-12 .OR. ICOMP.GT.NCOMPS) GO TO 633
        IF(ICOMP.GT.0) GO TO 209
        IF(ICOMP.EQ.0) GO TO 537
        IGO=-1*ICOMP
        GO TO (501,505,509,513,517,521,525,529,533,537,541,545),IGO
633   WRITE(*,*) BEEP,BEEP
      GO TO 533
537   ICOMP=0
      WRITE(*,737) IMIX
      WRITE(*,738)
      WRITE(*,739) I
      READ(*,*,ERR=637,END=637) ICOMP
        IF(ICOMP.LT.-12 .OR. ICOMP.GT.NCOMPS) GO TO 637
        IF(ICOMP.GT.0) GO TO 209
        IF(ICOMP.EQ.0) GO TO 541
        IGO=-1*ICOMP
        GO TO (501,505,509,513,517,521,525,529,533,537,541,545),IGO
637   WRITE(*,*) BEEP,BEEP
      GO TO 537
541   ICOMP=0
      WRITE(*,741) IMIX
      WRITE(*,742)
      WRITE(*,743) I
      READ(*,*,ERR=641,END=641) ICOMP
        IF(ICOMP.LT.-12 .OR. ICOMP.GT.NCOMPS) GO TO 641
        IF(ICOMP.GT.0) GO TO 209
        IF(ICOMP.EQ.0) GO TO 545
        IGO=-1*ICOMP
        GO TO (501,505,509,513,517,521,525,529,533,537,541,545),IGO
641   WRITE(*,*) BEEP,BEEP
      GO TO 541
545   ICOMP=0
      WRITE(*,745) IMIX
      WRITE(*,746)
```

```
          WRITE(*,747) I
          READ(*,*,ERR=645,END=645) ICOMP
            IF(ICOMP.LT.-12 .OR. ICOMP.GT.NCOMPS) GO TO 645
            IF(ICOMP.GT.0) GO TO 209
            IF(ICOMP.EQ.0) GO TO 501
            IGO=-1*ICOMP
            GO TO (501,505,509,513,517,521,525,529,533,537,541,545),IGO
  645     WRITE(*,*) BEEP,BEEP
          GO TO 545
  209     CONTINUE
C    ICOMP now contains component id of the Ith comp of mixture IMIX
          MIXCOM(IMIX,I)=ICOMP
          NCPM(IMIX,NCOMP)=NCOMP
          NCM(IMIX)=NCOMP
  108     CONTINUE
  106     CONTINUE
C         This section will build the component array as well as the
C             flag matrix which indicates which interaction parameters
C             will be needed.
C
     NC=0
     DO 211 I=1,NMIX
       DO 211 J=1,MXCMMX
         IF(MIXCOM(I,J).EQ.0) GO TO 211
         DO 210 K=1,NC
           IF(MIXCOM(I,J).EQ.NCID(K)) GO TO 211
  210      CONTINUE
         NC=NC+1
         IF(NC.GT.MAXCOM) WRITE(*,1003)
         NCID(NC)=MIXCOM(I,J)
  211    CONTINUE
C
C    Now alter contents of MIXCOM so as to refer comp # in overall C
     DO 213 I=1,NMIX
       DO 213 J=1,MXCMMX
         DO 212 K=1,NC
            IF(MIXCOM(I,J).EQ.0) GO TO 213
            IF(MIXCOM(I,J).NE.NCID(K)) GO TO 212
            MIXCOM(I,J)=ncid(K)
            GO TO 213
  212       CONTINUE
         WRITE(*,1002) I,J
         STOP
  213  CONTINUE
C
C    Now determine which interaction parameters are needed.
C
     DO 214 I=1,MAXCOM
       DO 214 J=1,MAXCOM
         IP(I,J)=0
```

```
   214     CONTINUE

       DO 240 IMIX=1,NMIX
         DO 235 I=1,MXCMMX
         IF(MIXCOM(IMIX,I).EQ.0) GO TO 235
         DO 230 J=I,MXCMMX
           IF(I.EQ.J) GO TO 230
           IF(MIXCOM(IMIX,J).EQ.0) GO TO 230
           DO 225 II=1,NC
           IF(MIXCOM(IMIX,I).NE.NCID(II)) GO TO 220
           I1=II

   220     IF(MIXCOM(IMIX,J).NE.NCID(II)) GO TO 225
           I2=II
   225     CONTINUE
           IP(I1,I2)=1
           IP(I2,I1)=1

   230     CONTINUE
   235     CONTINUE
   240     CONTINUE

C
C      Here ends input and starts calculations followed by output
C

      WRITE(*,2001) GOAL,NPROP,MODEL,NMIX,NC
      WRITE(*,2002) (I,NCID(I),I=1,NC)

      DO 3000 I=1,NMIX
        WRITE(*,2003) I,(MIXCOM(I,J),J=1,MXCMMX)
3000   CONTINUE

      WRITE(*,2004)
      DO 3001 I=1,NC
        WRITE(*,2005) (IP(I,J),J=1,NC)
3001   CONTINUE


        WRITE(*,*)' THE FOLLOWING PAIRS ARE THE PARAMETER REQUIRED'
        DO 3003 I=1,NC
        DO 3002 J=1,NC
        IF((IP(I,J).EQ.1)) THEN
        WRITE(*,*)NCID(I),'-',NCID(J)
        ELSE
        END IF
3002    CONTINUE
3003    CONTINUE

        WRITE(*,*)'ARE THE MATERIAL SELECTIONS CORRECT?'
        WRITE(*,*)' YES--->PRESS ANY KEY, NO--->PRESS N OR n'
        READ(*,3005)A
        IF (A.EQ.'N'.OR.A.EQ.'n') GO TO 205

        CALL SORT
        WRITE(*,*)' DO YOU WANT TO TRY ANOTHER ONE?'
        WRITE(*,*)' YES--->PRESS Y OR y, NO--->PRESS RETURN'
        READ(*,3005)A
        IF (A.EQ.'Y'.OR.A.EQ.'y') THEN
          GO TO 201
        END IF
3005    FORMAT(A)

      STOP
    1 FORMAT('1GOAL SELECTION MENU'/'0'/
     + '0Do you wish to :'/'0'/'0'/' 1)  Regress data'/
     + ' 2)  Predict properties'/' 3)  Control parameters'/
     + ' 4)  Quit program'/'0'/'0Enter choice --> '$)
    2 FORMAT('1PROPERTY SELECTION MENU'/'0'/'0Are you interested in :'/
     + '0'/'0'/' 1)  Vapor/Liquid Equilibrium (Isothermal P-x) Data'/
     + ' 2)  Vapor/Liquid Equilibrium (Isothermal T-x) Data'/
     + ' 3)  Heat-of-Mixing (Liquid Mixtures) Data'/
     + ' 4)  Liquid/Liquid Equilibrium Data'/
     + ' 5)  Return to Goal Selection Menu'/'0'/'0Enter choice --> '$)
    3 FORMAT('1MODEL SELECTION MENU #1'/'0'/
     + '0Which model would you like :'/'0'/'0'/
     + ' 1)  Knox Molecular (GAM) Model'/' 2)  Knox Group (GAG) Model'/
     + ' 3)  Knox Association Model'/
     + ' 4)  Knox Association Group Model'/' 5)  UNIQUAC Model'/
     + ' 6)  Modified UNIQUAC Model'/' 7)  UNIFAC Model'/
     + ' 8)  NRTL Model'/' 9)  Wilson Model'/' 0) Redlich-Kister Model'/
     + ' A)  Access Menu #2'/
     + ' B)  Return to Property Selection Menu'/'0'/
     + '0Enter choice --> '$)
    4 FORMAT('1MODEL SELECTION MENU #2'/'0'/
     + '0Which model would you like :'/'0'/'0'/
     + ' 1)  SRK Equation of State (Classical Mixing Rules)'/
     + ' 2)  SRK Equation of State (Knox Mixing Rules)'/
     + ' 3)  Peng-Robinson Equation of State'/
     + ' 4)  Future #1'/' 5)  Future #2'/' 6)  Future #3'/
     + ' 7)  Future #4'/' 8)  Future #5'/' 9)  Future #6'/
     + ' 0)  Future #7'/' A)  Access Menu #1'/
     + ' B)  Return to Property Selection Menu'/'0'/
     + '0Enter choice --> '$)
    5 FORMAT('1MIXTURE SELECTION MENU'/'0'/
     + '0How many mixtures do you have (or want to have) data on?'/
     + '0    An answer of zero (0) returns you to the'/
     + '          Model Selection Menu.'/
     + '0    The current maximum number of mixtures is ',I3,'.'/
     + '0'/'0Enter number --> '$)
    6 FORMAT('1NUMBER OF COMPONENTS FOR MIXTURE #',I3/'0'/
     + '0How many components are there in mixture #',I3,'?'/
     + '0    An answer of zero (0) returns you to the'/
     + '          Mixture Selection Menu.'/
     + '0    The current maximum number of components per'/
     + '          mixture is ',I3,'  and the overall maximum'/
     + '          number of components is ',I3,'.'/
     + '0'/'0Enter number --> '$)
  701 FORMAT('1COMPONENT SELECTION MENU FOR MIXTURE #',I3,',  PAGE 1'/
     + '0 1) Argon (Ar)                   21) Hydrogen Sulfide (H2S)'/
     + '  2) Boron Trichloride (BCl3)     22) Ammonia (NH3)'/
     + '  3) Boron Trifluoride (BF3)      23) Hydrazine (H4N2)'/
     + '  4) Bromine (Br2)                24) Helium-4 (He(4))'/
     + '  5) Nitrosyl Chloride (NOCl)     25) Iodine (I2)')
  702 FORMAT('  6) Chlorine (Cl2)               26) Krypton (Kr)'/
     + '  7) Phosphorus Trichloride (PCl3) 27) Nitric Oxide (NO)'/
     + '  8) Silicon Tetrachloride (SiCl4) 28) Nitrogen Dioxide (NO2)'/
     + '  9) Deuterium (D2)               29) Nitrogen (N2)'/
     + ' 10) Deuterium Oxide (D2O)        30) Nitrous Oxide (N2O)'/
     + ' 11) Fluorine (F2)                31) Neon (Ne)'/
     + ' 12) Nitr Trifluoride (NF3)       32) Oxygen (O2)'/
     + ' 13) Sil Tetrafluoride (SiF4)     33) Sulfur Dioxide (SO2)'/
     + ' 14) Sulf Hexafluoride (SF6)      34) Ozone (O3)'/
     + ' 15) Hydrogen Bromide (HBr)       35) Sulfur Trioxide (SO3)')
  703 FORMAT(' 16) Hydrogen Chloride (HCl)      36) Xenon (Xe)'/
     + ' 17) Hydrogen Fluoride (HF)       37) Trifl,bromomthne(CBrF3)'/
     + ' 18) Hydrogen Iodide (HI)         38) Chl,triflormthne(CClF3)'/
     + ' 19) Hydrogen (H2)                39) Dicl,difl,mthne(CCl2F2)'/
     + ' 20) Water (H2O)                  40) Phosgene (CCl2O)'/
     + ' < ) Return to Previous Page      > ) Go to Next Page'/
     + '0  Select component #',I3,' from the list above --> '$)
  705 FORMAT('1COMPONENT SELECTION MENU FOR MIXTURE #',I3,',  PAGE 2'/
     + '041) Tricl,fl,methane (CCl3F)   61) Methane (CH4)'/
     + ' 42) Carb Tet-chloride (CCl4)   62) Methanol (CH3OH)'/
     + ' 43) Carb Tet-fluoride (CF4)    63) Methyl Mercaptan (CH4S)'/
     + ' 44) Carbon Monoxide (CO)       64) Methyl Amine (CH3NH2)'/
     + ' 45) Carbonyl Sulfide (COS)     65) Methyl Hydrazine (CH6N2)'/
     + ' 46) Carbon Dioxide (CO2)       66) Cl,pentafl,ethane (C2CLF5)')
  706 FORMAT(' 47) Carbon Disulfide (CS2)   ',
     +                              ' 67) 11dicl,tetflethane(CCL2F4)'/
     + ' 48) Cl,diflormethane (CHClF2)  68) 12dicl,tetflethane(CCl2F4)'/
     + ' 49) Dicl,flormethane (CHCl2F)  69) 122tricltriflethne(CCl3F3)'/
     + ' 50) Chloroform (CHCl3)         70) Tetrachloroethylene(C2Cl4)'/
     + ' 51) Hydrogen Cyanide (HCN)     71) Tetcl12difethan (CCl4F2)'/
     + ' 52) Dibromomethane (CH2Br2)    72) Perfluoroethene (C2F4)'/
     + ' 53) Dichloromethane (CH2Cl2)   73) Perfluoroethane (C2F6)'/
     + ' 54) Formaldehyde (H2CO)        74) Cyanogen (C2N2)'/
     + ' 55) Formic Acid (HCOOH)        75) Trichlethylene (C2HCl3)')
  707 FORMAT(' 56) Methyl Bromide (CH3Br)    ',
     +                              ' 76) Trifl,acetic acid(CF3COOH)'/
    + ' 57) Methyl Chloride (CH3Cl)     77) Acetylene (C2H2)'/
    + ' 58) Methyl Fluoride (CH3F)      78) 11difluoroethylene(C2H2F2)'/
    + ' 59) Methyl Iodide (CH3I)        79) Ketene (C2H2O)'/
    + ' 60) Nitromethane (CH3NO2)       80) Vinyl Chloride (C2H3Cl)'/
    + ' < ) Return to Previous Page     > ) Go to Next Page'/
    + '0  Select component #',I3,' from the list above —> '$)
 709 FORMAT('1COMPONENT SELECTION MENU FOR MIXTURE #',I3,',  PAGE 3'/
     + '0 81) 1cl11difethane (CH3ClF2)  101) Dimethyl Ether ((CH3)2O)'/
     + '  82) Acetyl Chloride(C2H3ClO)  102) Ethanol (C2H5OH)'/
     + '  83) 112triclethane (C2H3Cl3)  103) EthyleneGlycol((CH2OH)2)'/
     + '  84) Vinyl Fluoride (C2H3F)    104) Ethyl Mercaptan (C2H6S)'/
     + '  85) 111-triflethane(C2H3F3)   105) DimethylSulfide((CH3)2S)'/
     + '  86) Acetonitrile (CH3CN)      106) Ethyl Amine (C2H5NH2)')
 710 FORMAT('  87) MethylIsocyanate (C2H3NO)',
     +                                 ' 107) Dimethyl Amine ((CH3)2NH)'/
     + '  88) Ethylene (C2H4)           108) Monoethanolamine (C2H7NO)'/
     + '  89) 1,1-dichlethane(C2H4Cl2)  109) Ethylenediamine (C2H8N2)'/
     + '  90) 1,2-dichlethane(C2H4Cl2)  110) Acrylonitrile (C3H3N)'/
     + '  91) 11difluoroethane(C2H4F2)  111) Propadiene (C3H4)'/
     + '  92) Acetaldehyde (CH3CHO)     112) MethylAcetylene (C3H4)'/
     + '  93) Ethylene Oxide (C2H4O)    113) Acrolein (C3H4O)'/
     + '  94) Acetic Acid (CH3COOH)     114) Acrylic Acid (C3H4O2)'/
     + '  95) Methyl Formate (HCOOCH3)  115) Vinyl Formate (C3H4O2)')
 711 FORMAT('  96) Ethyl Bromide (C2H5Br)   ',
     +                                 ' 116) Allyl Chloride (C3H5Cl)'/
     + '  97) Ethyl Chloride (C2H5Cl)   117) 123Trichlpropane(C3H5Cl3)'/
     + '  98) Ethyl Fluoride (C2H5F)    118) Propionitrile (C3H5N)'/
     + '  99) Ethylene Imine (C2H5N)    119) Cyclopropane (C3H6)'/
     + ' 100) Ethane (C2H6)             120) Propylene (C3H6)'/
     + '  < ) Return to Previous Page    > ) Go to Next Page'/
     + '0  Select component #',I3,' from the list above —> '$)
 713 FORMAT('1COMPONENT SELECTION MENU FOR MIXTURE #',I3,',  PAGE 4'/
     + '0121) 12-diclpropane (C3H6Cl2)  141) n-Propyl Amine (C3H9N)'/
     + ' 122) Acetone ((CH3)2CO)        142) Isopropyl Amine (C3H9N)'/
     + ' 123) Allyl Alcohol (C3H6O)     143) Trimethyl Amine (C3H9N)'/
     + ' 124) Propionaldehyde (C3H6O)   144) Maleic Anhydride(C4H2O3)'/
     + ' 125) Propylene Oxide (C3H6O)   145) Vinylacetylene (C4H4)'/
     + ' 126) VinylMethylEther (C3H6O)  146) Furan (C4H4O)')
 714 FORMAT(' 127) Propionic Acid (C3H6O2)  ,'
     +                                 ' 147) Thiophene (C4H4S)'/
     + ' 128) Ethyl Formate (C3H6O2)    148) Allyl Cyanide (C4H5N)'/
     + ' 129) Methyl Acetate (C3H6O2)   149) Pyrrole (C4H5N)'/
     + ' 130) Propyl Chloride (C3H7Cl)  150) 1-Butyne (C4H6)'/
     + ' 131) I-prpyl Chloride(C3H7Cl)  151) 2-Butyne (C4H6)'/
     + ' 132) Propane (C3H8)            152) 1,2-Butadiene (C4H6)'/
     + ' 133) 1-Propanol (C3H8O)        153) 1,3-Butadiene (C4H6)'/
     + ' 134) IsopropylAlcohol (C3H8O)  154) Vinyl Acetate (C4H6O2)'/
     + ' 135) MethylEthylEther (C3H8O)  155) AceticAnhydride (C4H6O3)')
 715 FORMAT(' 136) Methylal (C3H8O2)        ',
     +                                 ' 156) Dimethyl Oxalate (C4H6O4)'/
     + ' 137) 1,2-Propanediol (C3H8O2)  157) Succinic Acid (C4H6O4)'/
     + ' 138) 1,3-Propanediol (C3H8O2)  158) Butyronitrile (C4H7N)'/
     + ' 139) Glycerol (C3H8O3)         159) Methyl Acrylate (C4H7O2)'/
     + ' 140) MethylEthylSulfide(C3H8S) 160) 1-Butene (C4H8)'/
     + '  < ) Return to Previous Page     > ) Go to Next Page'/
     + '0  Select component #',I3,' from the list above --> '$)
 717 FORMAT('1COMPONENT SELECTION MENU FOR MIXTURE #',I3,',  PAGE 5'/
     + '0161) cis-2-Butene (C4H8)       181) n-Butane (C4H10)'/
     + ' 162) trans-2-Butene (C4H8)     182) Isobutane (C4H10)'/
     + ' 163) Cyclobutane (C4H8)        183) n-Butanol (C4H10O)'/
     + ' 164) Isobutylene (C4H8)        184) 2-Butanol (C4H10O)'/
     + ' 165) n-Butyraldehyde (C4H8O)   185) Isobutanol (C4H10O)'/
     + ' 166) Isobutyraldehyde (C4H8O)  186) tert-Butanol (C4H10O)')
 718 FORMAT(' 167) MethylEthylKetone (C4H8O)',
     +                                 ' 187) Ethyl Ether (C4H10O)'/
     + ' 168) Tetrahydrofuran (C4H8O)   188) 12Dimethoxyethan(C4H10O2)'/
     + ' 169) Vinyl Ethyl Ether (C4H8O) 189) DiethyleneGlycol(C4H10O3)'/
     + ' 170) n-Butyric Acid (C4H8O2)   190) Diethyl Sulfide (C4H10S)'/
     + ' 171) 1,4-Dioxane (C4H8O2)      191) DiethylDisulfide(C4H10S2)'/
     + ' 172) Ethyl Acetate (C4H8O2)    192) n-Butyl Amine (C4H11N)'/
     + ' 173) Isobutyric Acid (C4H8O2)  193) Isobutyl Amine (C4H11N)'/
     + ' 174) MethylPropionate(C4H8O2)  194) Diethyl Amine (C4H11N)'/
     + ' 175) n-PropylFormate (C4H8O2)  195) Pyridine (C5H5N)')
 719 FORMAT(' 176) 1-Chlorobutane (C4H9Cl)  ',
     +                                 ' 196) Cyclopentene (C5H8)'/
     + ' 177) 2-Chlorobutane (C4H9Cl)   197) 1,2-Pentadiene (C5H8)'/
     + ' 178) ter-ButylChloride(C4H9Cl) 198) 1trans-3Pentadiene (C5H8)'/
     + ' 179) Pyrrolidine (C4H9N)       199) 1,4-Pentadiene (C5H8)'/
     + ' 180) Morpholine (C4H9NO)       200) 1-Pentyne (C5H8)'/
     + '  < ) Return to Previous Page     > ) Go to Next Page'/
     + '0  Select component #',I3,' from the list above --> '$)
 721 FORMAT('1COMPONENT SELECTION MENU FOR MIXTURE #',I3,',  PAGE 6'/
     + ' 201) 2Methyl-13Butadiene(C5H8) 221) MthylIsobutyrate(C5H10O2)'/
     + ' 202) 3Methyl-12Butadiene(C5H8) 222) Piperidine (C5H11N)'/
     + ' 203) Cyclopentanone (C5H8O)    223) n-Pentane (C5H12)'/
     + ' 204) Ethyl Acrylate (C5H8O2)   224) 2-Methyl Butane (C5H12)'/
     + ' 205) Cyclopentane (C5H10)      225) 2,2-DimthylPropane(C5H12)'/
     + ' 206) 1-Pentene (C5H10)         226) 1-Pentanol (C5H12O)')
 722 FORMAT(' 207) cis-2-Pentene (C5H10)    ',
     +                                 ' 227) 2Methyl-1Butanol (C5H12O)'/
     + ' 208) trans-2-Pentene (C5H10)   228) 3Methyl-1Butanol (C5H12O)'/
     + ' 209) 2-Methyl-1-Butene (C5H10) 229) 2Methyl-2Butanol (C5H12O)'/
     + ' 210) 2-Methyl-2-Butene (C5H10) 230) 22Dimthyl1Propnol(C5H12O)'/
     + ' 211) 3-Methyl-1-Butene (C5H10) 231) EthylPropylEther (C5H12O)'/
     + ' 212) Valeraldehyde (C5H10O)    232) Perfluorobenzene (C6F6)'/
     + ' 213) MthylnPrpylKetone(C5H10O) 233) Perfluorocyclhxane(C6F12)'/
     + ' 214) MthyliPrpylKetone(C5H10O) 234) Perfluoro-n-hexane(C6F14)'/
     + ' 215) Diethyl Ketone (C5H10O)   235) o-Dichlorobnzene(C6H4Cl2)')
 723 FORMAT(' 216) n-Valeric Acid (C5H10O2) ',
     +                                 ' 236) m-Dichlorobnzene(C6H4Cl2)'/
     + ' 217) IsobutylFormate (C5H10O2) 237) p-Dichlorobnzene(C6H4Cl2)'/
     + ' 218) n-PropylAcetate (C5H10O2) 238) Bromobenzene (C6H5Br)'/
     + ' 219) EthylPropionate (C5H10O2) 239) Chlorobenzene (C6H5Cl)'/
     + ' 220) Methyl Butyrate (C5H10O2) 240) Fluorobenzene (C6H5F)'/
     + '  < ) Return to Previous Page    > ) Go to Next Page'/
     + '0  Select component #',I3,' from the list above --> '$)
 725 FORMAT('1COMPONENT SELECTION MENU FOR MIXTURE #',I3,',  PAGE 7'/
     + ' 241) Iodobenzene (C6H5I)        261) 23Dimethyl-1Butene(C6H12)'/
     + ' 242) Benzene (C6H6)             262) 23Dimethyl-2Butene(C6H12)'/
     + ' 243) Phenol (C6H6O)             263) 33Dimethyl-1Butene(C6H12)'/
     + ' 244) Aniline (C6H7N)            264) Cyclohexanol (C6H12O)'/
     + ' 245) 4-Methyl Pyridine (C6H7N) 265) MthylIsobtylKeton(C6H12O)'/
     + ' 246) 1,5-Hexadiene (C6H10)      266) n-Butyl Acetate (C6H12O2)')
 726 FORMAT(' 247) Cyclohexene (C6H10)       ',
     +                                  ' 267) IsobutylAcetate (C6H12O2)'/
     + ' 248) Cyclohexanone (C6H10O)     268) Ethyl Butyrate (C6H12O2)'/
     + ' 249) Cyclohexane (C6H12)        269) EthylIsobutyrate(C6H12O2)'/
     + ' 250) Methylcyclopentane(C6H12) 270) nPropylPropionat(C6H12O2)'/
     + ' 251) 1-Hexene (C6H12)           271) n-Hexane (C6H14)'/
     + ' 252) cis-2-Hexene (C6H12)       272) 2-Methylpentane (C6H14)'/
     + ' 253) trans-2-Hexene (C6H12)     273) 3-Methylpentane (C6H14)'/
     + ' 254) cis-3-Hexene (C6H12)       274) 2,2-Dimethylbutane(C6H14)'/
     + ' 255) trans-3-Hexene (C6H12)     275) 2,3-Dimethylbutane(C6H14)')
 727 FORMAT(' 256) 2-Methyl-2-Pentene(C6H12) 276) 1-Hexanol (C6H14O)'/
     + ' 257) 3Mthyl-cis2Pentene(C6H12) 277) EthylButylEther (C6H14O)'/
     + ' 258) 3Mthyl-trs2Pentene(C6H12) 278) DiisopropylEther (C6H14O)'/
     + ' 259) 4Mthyl-cis2Pentene(C6H12) 279) Dipropylamine (C6H15N)'/
     + ' 260) 4Mthyl-trs2Pentene(C6H12) 280) Triethylamine (C6H15N)'/
     + '  < ) Return to Previous Page    > ) Go to Next Page'/
     + '0  Select component #',I3,' from the list above --> '$)
 729 FORMAT('1COMPONENT SELECTION MENU FOR MIXTURE #',I3,',  PAGE 8'/
     + ' 281) Prflrmthylcychexan(C7F14) 301) 11Dimthylcycpentan(C7H14)'/
     + ' 282) Perfluoro-nheptane(C7F16) 302) cs12Dimthylcypntan(C7H14)'/
     + ' 283) Benzonitrile (C7H5N)      303) tr12Dimthylcypntan(C7H14)'/
     + ' 284) Benzaldehyde (C7H6O)      304) Ethylcyclopentane(C7H14)'/
     + ' 285) Benzoic Acid (C7H6O2)     305) Methylcyclohexane(C7H14)'/
     + ' 286) Toluene (C7H8)            306) 1-Heptene (C7H14)')
 730 FORMAT(' 287) MethylPhenylEther (C7H8O)',
     +                                 ' 307) 233Trimthyl1butane(C7H14)'/
     + ' 288) Benzyl Alcohol (C7H8O)    308) n-Heptane (C7H14)'/
     + ' 289) O-Cresol (C7H8O)          309) 2-Methylhexane (C7H16)'/
     + ' 290) m-Cresol (C7H8O)          310) 3-Methylhexane (C7H14)'/
     + ' 291) p-Cresol (C7H8O)          311) 22-Dimethylpentane(C7H16)'/
     + ' 292) 23Dimethylpyridine(C7H9N) 312) 23-Dimethylpentane(C7H16)'/
     + ' 293) 25Dimethylpyridine(C7H9N) 313) 24-Dimethylpentane(C7H16)'/
     + ' 294) 34Dimethylpyridine(C7H9N) 314) 33-Dimethylpentane(C7H16)'/
     + ' 295) 35Dimethylpyridine(C7H9N) 315) 3-Ethylpentane (C7H16)')
 731 FORMAT(' 296) Methylphenylamine (C7H9N)',
     +                                 ' 316) 223Trimethylbutane C7H16)'/
     + ' 297) o-Toluidine (C7H9N)       317) 1-Heptanol (C7H16O)'/
     + ' 298) m-Toluidine (C7H9N)       318) PhthalicAnhydride(C8H4O3)'/
     + ' 299) p-Toluidine (C7H9N)       319) Styrene (C8H8)'/
     + ' 300) Cycloheptane (C7H14)      320) MethylPhenylKetone(C8H8O)'/
     + '  < ) Return to Previous Page    > ) Go to Next Page'/
     + '0  Select component #',I3,' from the list above --> '$)
 733 FORMAT('1COMPONENT SELECTION MENU FOR MIXTURE #',I3,',  PAGE 9'/
     + ' 321) Methyl Benzoate (C8H8O2)  341) tr13Dimthylcychxan(C8H16)'/
     + ' 322) o-Xylene (C8H10)          342) c-14Dimthylcychxan(C8H16)'/
     + ' 323) m-Xylene (C8H10)          343) tr14Dimthylcychxan(C8H16)'/
     + ' 324) p-Xylene (C8H10)          344) Ethylcyclohexane (C8H16)'/
     + ' 325) Ethylbenzene (C8H10)      345) 112Trimthylcypntan(C8H16)'/
     + ' 326) o-Ethylphenol (C8H10O)    346) 113Trimthylcypntan(C8H16)')
 734 FORMAT(' 327) m-Ethylphenol (C8H10O)   ',
     +                                 ' 347) cct124TrMtylCyPntn(C8H16)'/
     + ' 328) p-Ethylphenol (C8H10O)    348) ctc124TrMtylCyPntn(C8H16)'/
     + ' 329) Phenetole (C8H10O)        349) 1Mthyl1Etylcypntan(C8H16)'/
     + ' 330) 2,3-Xylenol (C8H10O)      350) nPropylcyclopentan(C8H16)'/
     + ' 331) 2,4-Xylenol (C8H10O)      351) Isoprpylcycpentane(C8H16)'/
     + ' 332) 2,5-Xylenol (C8H10O)      352) 1-Octene (C8H16)'/
     + ' 333) 2,6-Xylenol (C8H10O)      353) trans-2-Octene (C8H16)'/
     + ' 334) 3,4-Xylenol (C8H10O)      354) n-Octane (C8H18)'/
     + ' 335) 3,5-Xylenol (C8H10O)      355) 2-Methylheptane (C8H18)')
 735 FORMAT(' 336) NNDimethylaniline(C8H11N)',
     +                                 ' 356) 3-Methylheptane (C8H18)'/
     + ' 337) 11Dimthylcychexane(C8H16) 357) 4-Methylheptane (C8H18)'/
     + ' 338) cs12Dimthylcychxan(C8H16) 358) 22-Dimethylhexane (C8H18)'/
     + ' 339) tr12Dimthylcychxan(C8H16) 359) 23-Dimethylhexane (C8H18)'/
     + ' 340) cs13Dimthylcychxan(C8H16) 360) 24-Dimethylhexane (C8H18)'/
     + '  < ) Return to Previous Page    > ) Go to Next Page'/
     + '0  Select component #',I3,' from the list above --> '$)
 737 FORMAT('1COMPONENT SELECTION MENU FOR MIXTURE #',I3,',  PAGE 10'/
     + ' 361) 25-Dimethylhexane (C8H18) 381) 1Mthyl3ethylbenzen(C9H12)'/
     + ' 362) 33-Dimethylhexane (C8H18) 382) 1Mthyl4ethylbenzen(C9H12)'/
     + ' 363) 34-Dimethylhexane (C8H18) 383) 123Trimthylbenzene(C9H12)'/
     + ' 364) 3-Ethylhexane (C8H18)     384) 124Trimthylbenzene(C9H12)'/
     + ' 365) 223Trimthylpentane(C8H18) 385) 135Trimthylbenzene(C9H12)'/
     + ' 366) 224Trimthylpentane(C8H18) 386) nPropylcyclohexane(C9H18)')
 738 FORMAT(' 367) 233Trimthylpentane(C8H18)',
     +                                 ' 387) Isoprpylcyclohexan(C9H18)'/
     + ' 368) 234Trimthylpentane(C8H18) 388) 1-Nonene (C9H18)'/
     + ' 369) 2Mthyl3ethylpentan(C8H18) 389) n-Nonane (C9H20)'/
     + ' 370) 3Mthyl3ethylpentan(C8H18) 390) 223Trimethylhexane(C9H20)'/
     + ' 371) 1-Octanol (C8H18O)        391) 224Trimethylhexane(C9H20)'/
     + ' 372) 2-Octanol (C8H18O)        392) 225Trimethylhexane(C9H20)'/
     + ' 373) 2-Ethylhexanol (C8H18O)   393) 33-Diethylpentane (C9H20)'/
     + ' 374) Butyl Ether (C8H18O)      394) 2233Tetramthylpntn(C9H20)'/
     + ' 375) Dibutylamine (C8H19N)     395) 2234Tetramthylpntn(C9H20)')
 739 FORMAT(' 376) alphaMethylStyrene(C9H10)',
     +                                 ' 396) 2244Tetramthylpntn(C9H20)'/
     + ' 377) Ethyl Benzoate (C9H10O2)  397) 2334Tetramthylpntn(C9H20)'/
```

```
   + ' 378) n-Propylbenzene (C9H12)   398) Naphthalene (C10H8)'/
   + ' 379) Isopropylbenzene (C9H12)  399) 1234Tethydnphthln(C10H12)'/

   + ' 380) 1Mthyl2Ethylbenzen(C9H12) 400) n-Butylbenzene (C10H14)'/
   + '  < ) Return to Previous Page    > ) Go to Next Page'/
   + '0  Select component #',I3,' from the list above --> '$)
741 FORMAT('1COMPONENT SELECTION MENU FOR MIXTURE #',I3,',  PAGE 11'/
   + ' 401) Isobutylbenzene (C10H14)  421) 2255Tetramthylhxn(C10H22)'/
   + ' 402) sec-Butylbenzene (C10H14) 422) 1-Decanol (C10H22O)'/
   + ' 403) tert-Butylbenzene(C10H14) 423) 1Mthylnaphthalene(C11H10)'/
   + ' 404) 1Mthl2isprpylbnzn(C10H14) 424) 2Mthylnaphthalene(C11H10)'/
   + ' 405) 1Mthl3isprpylbnzn(C10H14) 425) Butyl Benzoate (C11H14O2)'/
   + ' 406) 1Mthl4isprpylbnzn(C10H14) 426) nHexylcycpentane (C11H22)')
742 FORMAT(' 407) 14-Diethylbenzene(C10H14)',
   +                                 ' 427) 1-Undecene (C11H22)'/
   + ' 408) 1245Tetramthlbnzn(C10H14) 428) n-Undecane (C11H24)'/
   + ' 409) n-Butylaniline (C10H15N)  429) Diphenyl (C12H10)'/
   + ' 410) cis-Decalin (C10H18)      430) Diphenyl Ether (C12H10O)'/
   + ' 411) trans-Decalin (C10H18)    431) nHptylcyclpentane(C12H24)'/
   + ' 412) Caprylonitrile (C10H19N)  432) 1-Dodecene (C12H24)'/
   + ' 413) nButylcyclohexane(C10H20) 433) n-Dodecane (C12H26)'/
   + ' 414) Isobutylcychexane(C10H20) 434) Dihexyl Ether (C12H26O)'/
   + ' 415) secButylcychexane(C10H20) 435) Dodecanol (C12H26O)')
743 FORMAT(' 416) terButylcychexane(C10H20)',
   +                                 ' 436) Tributylamine (C12H27N)'/
   + ' 417) 1-Decene (C10H20)         437) Diphenylmethane (C13H12)'/
   + ' 418) n-Decane (C10H22)         438) nOctylcyclpentane(C13H26)'/
   + ' 419) 335Trimthylhptane(C10H22) 439) 1-Tridecene (C13H26)'/
   + ' 420) 2233Tetramthylhxn(C10H22) 440) n-Tridecane (C13H28)'/
   + '  < ) Return to Previous Page    > ) Go to Next Page'/
   + '0  Select component #',I3,' from the list above --> '$)
745 FORMAT('1COMPONENT SELECTION MENU FOR MIXTURE #',I3,',  PAGE 12'/
   + ' 441) Anthracene (C14H10)       461) n-Octadecene (C18H38)'/
   + ' 442) Phenanthrene (C14H10)     462) 1-Octadecanol (C18H38O)'/
   + ' 443) nNonylcyclpentane(C14H28) 463) nTetrdecylcycpntn(C19H38)'/
   + ' 444) 1-Tetradecene (C14H28)    464) n-Nonadecane (C19H40)'/
   + ' 445) n-Tetradecane (C14H30)    465) nPntadecylcycpntn(C20H40)'/
   + ' 446) nDecylcyclpentane(C15H30) 466) n-Eicosane (C20H42)')
746 FORMAT(' 447) 1-Pentadecene (C15H30)   ',
   +                                 ' 467) 1-Eicosanol (C20H42O)'/
   + ' 448) n-Pentadecane (C15H32)    468) nHexadecylcycpntn(C21H42)'/
   + ' 449) Dibutl-o-pthlat(C16H22O4)'/
   + ' 450) nDecylcyclohexane(C16H32)'/
   + ' 451) 1-Hexadecene (C16H32)'/
   + ' 452) n-Hexadecane (C16H34)'/
   + ' 453) nDodecylcycpentan(C17H34)'/
   + ' 454) Heptadecanol (C17H36O)'/
   + ' 455) n-Heptadecane (C17H36)')
747 FORMAT(' 456) o-Terphenyl (C18H14)'/
   + ' 457) m-Terphenyl (C18H14)'/
```

```
     + ' 458) p-Terphenyl (C18H14)'/
     + ' 459) 1-Octadecene (C18H36)'/
     + ' 460) nTridecylcyclpntn(C18H36)'/
     + '  < ) Return to Previous Page     > ) Go to Next Page'/
     + '0  Select component #',I3,' from the list above —> '$)
   11 FORMAT(A1)
   12 FORMAT(A3)
 1001 FORMAT('0',A1,'You have entered an illegal response.',A1)
 1002 FORMAT('0ERROR!! Problem in setup of component matrices.')
 1003 FORMAT('0You have exceeded the maximum number of components.')
 2001 FORMAT('1  Goal  = ',I4/'   Prop  = ',I4/
     +      '   Model = ',I4/'   Mixes = ',I4/'   Comps = ',I4)
 2002 FORMAT('1  The following components have been selected:'/' '/
     +'  ',20(I6,')',' ',I6))

 2003 FORMAT('0'/'0Mixture # ',I2,' contains choices ',10I6)
 2004 FORMAT('1 The following is the parameter required matrix.'/'0')
 2005 FORMAT(' ',20I3)
      END
```

```
          SUBROUTINE SORT
C         THIS IS THE SUBROUTINE TO CLASSIFY THE SELECTION MENU
C
          INTEGER GOAL
          COMMON/CNTRL/MODEL,GOAL,NPROP,NC,NG
          GO TO (5,300,300,300,5,5,300,5,5,5,
     +    300,300,300,300,300,300,300,300,300,300) MODEL
 5        GO TO (10,20,30) GOAL
10        GO TO 300

20        GO TO (40,50,60,70)NPROP


30        CALL PARA
          RETURN
40        CALL DEWP
          RETURN
50        CALL DEWT
          RETURN
60        CALL MIXHE
          RETURN
70        CALL ELIPS
          RETURN
300       WRITE(*,*)' SORRY! PROGRAM IS NOT COMPLETED'
310       RETURN
          END
```

```
      SUBROUTINE DEWP

      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER GOAL
      COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     + SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     + OMEGA(20),F(20)
      COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX,
     + MIXCOM(50,10),NCM(10)
      COMMON/CNTRL/MODEL,GOAL,NPROP,NC,NG
      COMMON/RST/HENRY(10,30),CHENRY(10,30,2),FREFER(20),
     + ZRA(20),VLIQ(20),VLIQL(20,20),CVLIQL(20,20,2)
      COMMON/VIRIAL/B(20,20),BD(20,20),BF(20,20),DB(20,20),
     + DBD(20,20),ETA(20),DIPOLE(20),ET(20,20)
      COMMON/PHIS/PHI(20),PHL(20),BMIX
      COMMON/ACTCO/CACTCO(20,20,6),GAMMA(20),HERT
      CHARACTER*40 NAME(20)
      CHARACTER*1 A

C     DATA INPUT

      DO 400 IMIX=1,NMIX
      NCOMP=NCM(IMIX)

      CALL INPUT1

    1 WRITE(*,5)IMIX
    5 FORMAT(1X,' READ SYSTEM TEMPERATURE OF MIXTURE #',I2)
   10 READ(*,*)T
      DO 25 I=1,NCOMP
      WRITE(*,20)NAME(I)
   20 FORMAT(1X,'READ VAPOR COMPOSITION OF ',A40)
      READ(*,*)Y(I)
   25 CONTINUE
      WRITE(*,*)' ARE YOU SATISFIED FOR THE INPUTTED DATA?'
      WRITE(*,*)'Yes------>Press y or Y,No----->Press N or n'
      READ(*,390)A1
      IF(A1.EQ.'N'.OR.A1.EQ.'n') GO TO 1

      IF ((T.LT.0).OR.(Y(I).LT.0)) GO TO 1


      Pi=1.0
      SUMY=0.0

      DO 100 I=1,NCOMP
      GAMMA(I)=1.0
```

```
          X(I)=1.0/NCOMP
          SUMY=SUMY+Y(I)
100       CONTINUE

          RT=82.057*T
          P=Pi

105       CALL VIRIAL


C         LIQUID PHASE ACTIVITY COEFFICIENTS.

          CALL PHIMIX

          CALL RSTATE

          SUMX=0.0

          DO 110 I=1,NCOMP
          F(I)=Y(I)*PHI(I)*P
          X(I)=F(I)/(GAMMA(I)*FREFER(I)*DEXP(VLIQ(I)*P/RT))
          SUMX=SUMX+X(I)

110       CONTINUE

          SUMXO=SUMX
113       DO 114 I=1,NCOMP
          X(I)=X(I)/SUMX
114       CONTINUE

          CALL ACTCO

145       CALL RSTATE

          SUMX=0.0

          DO 150 I=1,NCOMP


          X(I)=F(I)/(GAMMA(I)*FREFER(I)*DEXP(VLIQ(I)*P/RT))


          SUMX=SUMX+X(I)
150       CONTINUE

          IF (ABS(SUMXO-SUMX)/SUMX.GT.1.0E-5) THEN
            SUMXO=SUMX
            GO TO 113
          END IF
```

```
C     CHECK TO SEE IF SUMX IS UNITY

170   SUMX1=SUMX-1.0

      IF (ABS(SUMX1).LT.1.0E-5) GO TO 300

      P=P/SUMX

      IF (P.LT.1.E-7.OR.P.GT.150.) RETURN
      GO TO 105
300   WRITE(*,310)IMIX,P
310   FORMAT(1X,'THE SYSTEM PRESSURE OF MIXTURE #',I2,' IS',1x,D10.4,
     + ' atm')

      DO 360 I=1,NCOMP

      WRITE(*,350)NAME(I),X(I)
350   FORMAT(1X,'LIQUID COMPOSITION OF',1x,A40,'IS',/1X,D10.4)
360   CONTINUE

      WRITE(*,380)
380   FORMAT(1X,'ARE YOU SATISFIED WITH THE RESULT?'/1X,'IF SATISFIED,
     + PRESS Y OR NOT, PRESS N')
      READ(*,390)A
390   FORMAT(A)
      IF (A.EQ.'N'.OR.A.EQ.'n') GO TO 1
400   CONTINUE
410   RETURN
      END
```

```
              SUBROUTINE DEWT
C       SUBROUTINE PROGRAM FOR CALCULATION OF TEMPERATURE AND LIQUID
C       COMPOSITION FROM VAPOR COMPOSITON AND TOTALPRESSURE DATA
C
C       THIS PROGRAM REQUIRES THE FOLLOWING SUBROUTINES— INPUT1,VIRIAL,
C       RSTATE,ACTCO,PHIMIX

        IMPLICIT REAL*8(A-H,O-Z)
        INTEGER GOAL

        COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     +  SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     +  OMEGA(20),F(20)
        COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX,
     +  MIXCOM(50,10),NCM(10)
        COMMON/CNTRL/MODEL,GOAL,NPROP,NC,NG
        COMMON/RST/HENRY(10,30),CHENRY(10,30,2),FREFER(20),
     +  ZRA(20),VLIQ(20),VLIQL(20,20),CVLIQL(20,20,2)
        COMMON/VIRIAL/B(20,20),BD(20,20),BF(20,20),DB(20,20),
     +  DBD(20,20),ETA(20),DIPOLE(20),ET(20,20)
        COMMON/PHIS/PHI(20),PHL(20),BMIX
        COMMON/ACTCO/CACTCO(20,20,6),GAMMA(20),HERT
        CHARACTER*40 NAME(20)
        CHARACTER*1 A


C       DATA INPUT
        DO 400 IMIX=1,NMIX

        NCOMP=NCM(IMIX)
        CALL INPUT1
1       WRITE(*,5)IMIX
5       FORMAT(1X,'READ SYSTEM PRESSURE OF MIXTURE #',I2)
10      READ(*,*)P
        DO 25 I=1,NCOMP
        WRITE(*,20)NAME(I)
20      FORMAT(1X,'READ VAPOR COMPOSITION OF ',A40)
        READ(*,*)Y(I)
25      CONTINUE
        WRITE(*,*)' ARE YOU SATISFIED FOR THE INPUTTED DATA?'
        WRITE(*,*)'Yes——>Press y or Y,No——>Press N or n'
        READ(*,390)A1
        IF(A1.EQ.'N'.OR.A1.EQ.'n') GO TO 1

c       GUESS INITIAL TEMPERATURE FOR FIRST ITERATION


        NTINT=(NCOMP+NLIGHT+1)/2
        T=0.7*TC(NTINT,NTINT)
        SUMY=0.0
        DO 30 I=1,NCOMP
        GAMMA(I)=1.0
        X(I)=1.0/NCOMP
        SUMY=SUMY+Y(I)

30      CONTINUE

        IF((P.LT.0).OR.(Y(I).LT.0)) GO TO 1

C       BEGIN A TEMPERATURE ITERATION
C

        L=1
50      RT=82.057*T

        CALL VIRIAL
        CALL RSTATE
        CALL PHIMIX
60      SUMX=0.0

        DO 70 I=1,NCOMP
        F(I)=Y(I)*PHI(I)*P
        X(I)=F(I)/(GAMMA(I)*FREFER(I)*EXP(VLIQ(I)*P/RT))
        SUMX=SUMX+X(I)

70      CONTINUE
        SUMXO=SUMX

*       NORMALIZE LIQUID COMPOSITON

72      DO 75 I=1,NCOMP
75      X(I)=X(I)/SUMX

C       BEGIN AN ITERATION TO FIND LIQUID PHASE ACTIVTY COEFFICIENTS

80      CALL ACTCO
        TOLD=T
        SUMX=0.0
        DO 90 I=1,NCOMP
        X(I)=F(I)/(GAMMA(I)*FREFER(I)*EXP(VLIQ(I)*P/RT))
        SUMX=SUMX+X(I)
90      CONTINUE
C
C       CHECK TO SEE WHETHER SUMX IS CONSTANT
C
        IF (ABS(SUMX-SUMXO)/SUMX-1.0E-5) 110,110,100
100     SUMXO=SUMX
        CALL RSTATE
```

```
      GO TO 72

C
C     CHECK TO SEE WHETHER SUMX IS UNITY
C
110   SUMX1=SUMX-1.0
      IF (ABS(SUMX1).LT.1.0E-5) GO TO 300

C
C     ADJUST THE TEMPERATURE AND START ANOTHER ITERATION
C
C
140   IF (SUMX1) 160,300,150
150   T=T*(1.0+0.02/L)
      L=L+1
      GO TO 50
160   T=T*(1.0-0.02/L)
      L=L+1
      GO TO 50
300   WRITE(*,310)IMIX,T
310   FORMAT(1X,'THE SYSTEM TEMPERATURE OF MIXTURE #',I2,'IS',
     $ 1X,D10.4,' k')
      DO 360  I=1,NCOMP
      WRITE(*,350)NAME(I),X(I)
350   FORMAT(1X,'LIQUID COMPOSITION OF',1X,A40,'IS',/1X,D10.4)
360   CONTINUE
      WRITE(*,380)
380   FORMAT(1X,'ARE YOU SATISFIED WITH THE RESULT?'/1X,'IF SATISFIED,
     $ PRESS Y OR NOT,PRESS N')
      READ(*,390)A
390   FORMAT(A)
      IF (A.EQ.'N'.OR.A.EQ.'n') GO TO 1
400   CONTINUE
410   RETURN
      END
```

```
        SUBROUTINE MIXHE
*
* THIS SUBROUTINE CALCULATES HEAT OF MIXING IN LIQUID PHASE. THE REQUI-
* RED SUBROUTINES ARE FROM ACTCO1 TO ACTCO20.
*

        IMPLICIT REAL*8(A-H,O-Z)
        INTEGER GOAL
        COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     +  SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     +  OMEGA(20),F(20)
        COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX,
     +  MIXCOM(50,10),NCM(10)
        COMMON/CNTRL/MODEL,GOAL,NPROP,NC,NG
        COMMON/RST/HENRY(10,30),CHENRY(10,30,2),FREFER(20),
     +  ZRA(20),VLIQ(20),VLIQL(20,20),CVLIQL(20,20,2)
        COMMON/VIRIAL/B(20,20),BD(20,20),BF(20,20),DB(20,20),
     +  DBD(20,20),ETA(20),DIPOLE(20),ET(20,20)
        COMMON/PHIS/PHI(20),PHL(20),BMIX
        COMMON/ACTCO/CACTCO(20,20,6),GAMMA(20),HERT
        CHARACTER*40 NAME(20)
        CHARACTER*1 A
        DATA R/1.987/,R1/8.314/
        DO 800 IMIX=1,NMIX
        NCOMP=NCM(IMIX)

* INPUT  OF SYSTEM CONDITION

        CALL INPUT1

1       WRITE(*,5) IMIX
5       FORMAT(1X,'READ SYSTEM TEMPERATURE OF MIXTURE #',I2)
        READ(*,*) T
        RT=R*T
        R1T=R1*T
        DO 10 I=1,NCOMP
          WRITE(*,8) NAME(I)
8         FORMAT(1X,'READ LIQUID COMPOSITION OF',2X,A40)
10        READ(*,*) X(I)
        WRITE(*,*)' ARE YOU SATTISFIED FOR THE INPUTTED DATA?'
        WRITE(*,*)'Yes------>Press y or Y,No---->Press N or n'
        READ(*,550)A1
        IF(A1.EQ.'N'.OR.A1.EQ.'n') GO TO 1

        CALL RSTATE
        CALL ACTCO

* OUT PUT
```

```
500     WRITE(*,505)IMIX
505     FORMAT(1X,'HEAT OF MIXING OF MIXTURE #',I2)
        WRITE(*,510)HERT*RT,HERT*R1T
510     FORMAT(1X,D10.4,' CAL/MOL OR ',D10.4,'J/MOL')

        WRITE(*,520)
520     FORMAT(1X,'ARE YOU SATISFIED WITH THE RESULT?'/1X,'IF SATISFIED,
     $  PRESS Y OR NOT,PRESS N')
        READ(*,550)A
550     FORMAT(A1)
        IF (A.EQ.'Y'.OR.A.EQ.'y') THEN
          GO TO 800
        ELSE
          GO TO 1
        END IF

800     CONTINUE
        RETURN
        END
        SUBROUTINE ELIPS
* ELIPS CALCULATES CONJUGATE PHASE COMPOSITIONS XR AND XE FOR PARTIALLY
* MISCIBLE N COMPONENT LIQUID SYSTEMS (N.LE.20) OF OVERALL COMPOSITION
* Z AT GIVEN TEMPERATURE T(K).IR AND IE ARE THE COMPONENT NRS OF THE R
* AND E PHASE SOLVENTS (SHOULD BE SET TO 0 IF NO SUCH SOLVENTS
* DESIGNATED).
* THE FRACTION OF THE SYSTEM GOING INTO THE E PHASE IS RETURNED IN A,
* WITH A BEING 0 OR 1 (AND XE=XR) IF Z IS IN A SINGLE PHASE REGION.

        IMPLICIT REAL*8(A-H,O-Z)
        INTEGER GOAL
        COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     +  SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     +  OMEGA(20),F1(20)
        COMMON/PARIP/NCID(20),LP(20,20),NCPM(50,10),NCOMP,NMIX,
     +  MIXCOM(50,10),NCM(10)
        COMMON/LILIK/XR(20),XE(20),AK(20),GAR(20),GAE(20)
        CHARACTER*40 NAME(20)
        CHARACTER*1 A1
        REAL Z(20),K1(20),KS,KP,K2,RX(20),EX(20)
*
* DATA INPUT:1.SYSTEM TEMPERATURE
*            2.FEED COMPOSITION
*

      DO 1000 IMIX=1,NMIX
        NCOMP=NCM(IMIX)
        CALL INPUT1
1       WRITE(*,5)IMIX
5       FORMAT(1X,'READ SYSTEM TEMPERATURE OF MIXTURE #',I2)
10      READ(*,*)T

        DO 25 I=1,NCOMP
          WRITE(*,20)NAME(I)
20        FORMAT(1X,' READ FEED COMPOSITION OF ',A40)
25        READ(*,*)Z(I)

        WRITE(*,*)' ARE YOU SATISFIED FOR THE INPUTTED DATA?'
        WRITE(*,*)'Yes---->Press y or Y,N---->Press N or n'
        READ(*,430)A1
        IF(A1.EQ.'N'.OR.A1.EQ.'n') GO TO 1

* FOR NEW SYSTEMS WITHOUT IR,IE SPECIFIED,FIND IR,IE AS LEAST SOLUBLE
* PAIR

101     KS=1.
        KP=1.
110     DO 115 I=1,NCOMP
        XR(I)=0.
115     XE(I)=0.
```

```
        J1=1
        IF(IR.EQ.0) THEN
          IF(IE.EQ.0) THEN
116         DO 119 J=2,NCOMP
              XR(J1)=0.
              XE(J1)=0.
              J1=J-1
              XR(J)=0.98
              XE(J)=0.02
              XR(J-1)=0.
              XE(J-1)=0.
              IF(Z(J).LT.0.10) GO TO 119
                DO 118 I=1,J1
                  XE(I)=0.98
                  XR(I)=0.02
                  IF(I.GT.1) THEN
                    XE(I-1)=0.
                    XR(I-1)=0.
                  END IF
                  IF(Z(I).LT.0.10) GO TO 118
                  CALL LILIK
                  IF(AK(I).GT.KS) THEN
                    KS=AK(I)
                    IS=I
                  ELSE
                    IF(1./AK(J).GT.KS) THEN
                      KS=1./AK(J)
                      IS=J
                    END IF
                  END IF
118             CONTINUE
119         CONTINUE
            XR(NCOMP)=0.
            XE(NCOMP)=0.
            XR(J1)=0.
            XE(J1)=0.
          ELSE
            IS=IE
          END IF
        ELSE
          IF(IE.EQ.0) THEN
            IS=IR
          ELSE
            GO TO 130
          END IF
        END IF

125     XE(IS)=0.98
        XR(IS)=0.02
126     DO 129 J=1,NCOMP
```

```
          IF(J.EQ.IS) GO TO 129
          XR(J)=0.98
          XE(J)=0.02
          IF(J.EQ.(IS+1)) THEN
            IF (J.GT.2) THEN
              XR(J-2)=0.
              XE(J-2)=0.
            END IF
          ELSE
            IF (J.GT.1) THEN
              XR(J-1)=0.
              XE(J-1)=0.
            END IF
          END IF
127       IF (Z(J).LT.0.10) GO TO 129
          CALL LILIK
          IF(AK(J).GE.KP) GO TO 129
          KP=AK(J)
          IP=J
129     CONTINUE
        IE=IP
        IF(IR.NE.IS) THEN
          IE=IS
          IR=IP
        END IF

* INITIALIZE R AND E PHASE COMPOSITIONS
130     DO 131 I=1,NCOMP
          XR(I)=0.
131       XE(I)=0.
        XR(IR)=0.98
        XE(IR)=0.02
        XR(IE)=0.02
        XE(IE)=0.98

* GET INITIAL ESTIMATES FOR AK VALUES
135     CALL LILIK
        SZ=0.
136     DO 137 I=1,NCOMP
          SZ=SZ+Z(I)
137       K1(I)=AK(I)-1.
        IF(ABS(SZ-1.0).GT.0.01) GO TO 903

* GET INITIAL ESTIMATE FOR A
        A=Z(IE)/(Z(IE)+Z(IR))
        A=-A/K1(IR)+(A-1.)/K1(IE)
        IF(A.LT.0.) A=0.
        IF(A.GT.1.) A=1.
        AO=A
        IT=0
        KAC=0
* CONDUCT ITERATION OVER PHASE COMPOSITION (OUTER LOOP)
200     IT=IT+1
        IF(IT.GT.50) GO TO 900
        AA=AO
        AO=A
* CONDUCT NEWTON-RAPHSON ITERATION FOR A AT FIXED AK VALUES
300     DO 319 M=1,10
          F=0.
          DF=0.
301     DO 309 I=1,NCOMP
          K2=A*K1(I)+1.
          F=F+Z(I)*K1(I)/K2
309       DF=DF-Z(I)*(K1(I)/K2)**2
        IF(ABS(F).LT.1.E-05) GO TO 210
        DA=F/DF
310     AN=A-DA
* LIMIT A TO RANGE 0-1 IF IT FALLS OUTSIDE POLES OF F.
        IF(AN.GE.-1./K1(IR)) THEN
          AN=A+(1./K1(IR)+A)/2.
        ELSE
          IF(AN.LE.-1./K1(IE)) THEN
            AN=A+(1./K1(IE)+A)/2.
          END IF
        END IF
        A=AN
319     CONTINUE
        GO TO 900
210     A=AN
        SR=0.
        SE=0.
* FIND NEW R AND E PHASE COMPOSITIONS
211     DO 215  I=1,NCOMP
          XR(I)=Z(I)/(A*K1(I)+1.)
          XE(I)=AK(I)*XR(I)
          SR=SR+XR(I)
215       SE=SE+XE(I)
220     DO 225  I=1,NCOMP
          XR(I)=XR(I)/SR
225       XE(I)=XE(I)/SE

* AT ALTERNATE ITERATIONS AFTER 3 ACCELERATE PHASE COMPOSITIONS BY
* WEGSTEIN METHOD BASED ON SUM OF DEVIATIONS OF COMPONENT FUGACITIES.
230     IF(IT.GE.3) THEN
          IF(IT.GT.5.OR.(A.GE.0..AND.A.LE.1.)) THEN
            IF(KAC.LT.1) THEN
              IF(SS.LE.SL) THEN
                IF(SS.LE.0.2) THEN
                  KAC=1
                  WG=SS/(SL-SS)
                  WK=1.+WG
                  SR=0.
                  SE=0.
                  DO  235     I=1,NCOMP
                    XR(I)=WK*XR(I)-WG*RX(I)
* ALLOW NO NEGATIVE MOL FRACTIONS
                    IF(XR(I).LT.0.) XR(I)=0.
                    XE(I)=WK*XE(I)-WG*EX(I)
                    IF(XE(I).LT.0.) XE(I)=0.
                    SR=SR+XR(I)
235                 SE=SE+XE(I)
* NORMALIZE ACCELERATED COMPOSITIONS (TO ALLOW FOR XR(I) OR XE(I) SET
* TO 0)
236               DO 237 I=1,NCOMP
                    XR(I)=XR(I)/SR
237                 XE(I)=XE(I)/SE
                END IF
              END IF
            ELSE
              KAC=0.
            END IF
          END IF
        END IF
* GET NEW AK AND GAMMA VALUES

250     CALL LILIK
        ESS=1.E-03
        IF(IT.GT.5) THEN
* CHECK FOR VICINITY OF A PLAIT POINT
          PPI=AK(IR)/AK(IE)+AK(IE)/AK(IR)
            IF(PPI.LE.10..AND.SS.LE.0.05) THEN

* EXIT IF TOO NEAR PLAIT POINT
              IF(IT.GE.20) GO TO 290
              IF(PPI.LE.7.) THEN
* CHECK IF CALCULATION NEAR PLAIT POINT IS PROBABLY IN SINGLE PHASE
* REGION--IF SO CONTINUE.
                DE=AO
                IF(DE.GT.0.5) DE=AO-1.
                DI=(AA-A)/DE
                IF(DI.LT.0.1) GO TO 290
              END IF
            END IF
          IF(PPI.LT.20) ESS=2.E-04
        END IF
        SL=SS
        SS=0.
253     DO 255 I=1,NCOMP
          K1(I)=AK(I)-1.
* CALCULATE OBJECTIVE FUNCTION
```

```
255       SS=SS+ABS(GAE(I)*XE(I)-GAR(I)*XR(I))
* CHECK CONVERGENCE
        IF(SS.GT.ESS) THEN
          IF(A.LT.0..OR.A.GT.1.) THEN
            IF(IT.GE.3.AND.SS.LE.0.20) THEN
              IF(IT.GE.5.OR.SS.LE.0.05) THEN
* CHECK IF A MOVING AWAY FROM 0-1 REGION
                IF((ABS(A)-ABS(AO)).GT.0.) GO TO 195
              END IF
            END IF
          END IF
        ELSE
          IF(KAC.NE.1) THEN
            GO TO 191
          END IF
        END IF
* SAVE LAST PHASE COMPOSITIONS FOR USE IN ACCELERATION
260     DO 265 I=1,NCOMP
          RX(I)=XR(I)
265       EX(I)=XE(I)
        GO TO 200
* FEED IN VICINITY OF PLAIT POINT--CHECK IF IN TWO PHASE REGION
290     IF(A.GE.0..AND.A.LE.1.) GO TO 905
        GO TO 195
* DO NOT ALLOW CONVERGENCE ON ACCELERATED ITERATION
* CONVERGED SOLUTION--CHECK IF IN TWO PHASE REGION
191     IF(A.LT.0..OR.A.GT.1.) THEN
* FEED OUTSIDE TWO PHASE REGION
195       DO 196 I=1,NCOMP
            XR(I)=Z(I)
196         XE(I)=Z(I)
          IF(A.LT.0.) A=0.
          IF(A.GT.1.) A=1.
        END IF
        WRITE(*,360)IMIX
360     FORMAT(1X,'LIQUID MIXTURE #',I2,'HAS FOLLOWING COMPOSITIONS')
        DO 380 I=1,NCOMP
380     WRITE(*,400)NAME(I),XE(I),XR(I)
400     FORMAT(1X,A40/' EXTRACTED PHASE:',10X,D10.4,2X,'RAFFINATE
     + PHASE:',D10.4)
        WRITE(*,420)
420     FORMAT(1X,'ARE YOU SATISFIED WITH THE RESULT?'/1X,'IF SATISFIED,
     +  PRESS Y OR NOT,PRESS N')
        READ(*,430)A1
430     FORMAT(A)
        IF (A1.EQ.'N'.OR.A1.EQ.'n') GO TO 1
        GO TO 1000
* ON FAILURE TO CONVERGE SET A TO -1
900     WRITE(*,*)'FAILURE TO CONVERGE'
        GO TO 910
```

```
* FOR BAD INPUT DATA SET A TO -1
903     WRITE(*,*)'BAD INPUT DATA'
        GO TO 910
* FOR FEED TOO NEAR PLAIT POINT SET A TO -1
905     WRITE(*,*)'FEED TOO NEAR PLAIT POINT'
910     DO 911 I=1,NCOMP
          XR(I)=Z(I)
911       XE(I)=Z(I)
          A=-1.
        GO TO 1
1000    CONTINUE
        RETURN
        END

        SUBROUTINE LILIK
* LILIK CALCULATES TWO PHASE EFFECTIVE LIQUID EQUILIBRIUM RATIOS K
* FOR ALL N COMPONENTS (N.LE.20) WHOSE INDICES APPEAR IN VECTOR ID,
* GIVEN TEMPERATURE T(K) AND ESTIMATES OF PHASE COMPOSITIONS XR AND XE
* (USED WITHOUT CORRECTION TO EVALUATE ACTIVITY  COEFFICIENTS GAR AND
* GAE).

        IMPLICIT REAL*8(A-H,O-Z)
        COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     +  SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     +  OMEGA(20),F(20)
        COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX,
     +  MIXCOM(50,10),NCM(10)
        COMMON/ACTCO/CACTCO(20,20,6),GAMMA(20),HERT
        COMMON/LILIK/XR(20),XE(20),AK(20),GAR(20),GAE(20)
        CHARACTER*40 NAME(20)

* CONVERT COMPOSITION VECTORS TO DIMENSION 20 TO MATCH LOWER LEVEL
* SUBROUTINES
        IF (MODEL.EQ.9) THEN
          WRITE(*,*)'WILSON MODEL CAN NOT BE USED'
          RETURN
        END IF
* CALCULATE RAFFINATE PHASE ACTIVITY COEFFICIENTS
101     DO 102 I=1,NCOMP
          X(I)=XR(I)
102     CONTINUE
        CALL ACTCO
120     DO 125 I=1,NCOMP
125       GAR(I)=GAMMA(I)


* CALCULATE EXTRACT PHASE ACTIVITY COEFFICIENTS
        DO 140 I=1,NCOMP
140       X(I)=XE(I)
        CALL ACTCO
```

```
      DO 150 I=1,NCOMP
150     GAE(I)=GAMMA(I)
* GET K VALUE
      DO 200 I=1,NCOMP

        AK(I)=GAR(I)/GAE(I)

200     IF (AK(I).LE.0..OR.AK(I).GT.1.E+19) GO TO 900
      RETURN
* FOR A K VALUE OUT OF RANGE SET ERR TO 2

900   DO 905 I=1,NCOMP
905   AK(I)=0.
      RETURN
      END
      SUBROUTINE INPUT1

      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER GOAL
      COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     + SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     + OMEGA(20),F(20)
      COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX,
     + MIXCOM(50,10),NCM(10)
      COMMON/CNTRL/MODEL,GOAL,NPROP,NC,NG
      COMMON/RST/HENRY(10,30),CHENRY(10,30,2),FREFER(20),
     + ZRA(20),VLIQ(20),VLIQL(20,20),CVLIQL(20,20,2),CVLIQ(20,3)
      COMMON/VIRIAL/B(20,20),BD(20,20),BF(20,20),DB(20,20),
     + DBD(20,20),ETA(20),DIPOLE(20),ET(20,20)
      COMMON/PHIS/PHI(20),PHL(20),BMIX
      COMMON/ACTCO/CACTCO(20,20,6),GAMMA(20),HERT
      COMMON/PSAT/ANTA(20),ANTB(20),ANTC(20),PSAT(20)
      COMMON/ACTCOU/R(20),Q(20),QP(20)

      CHARACTER*40 NAME(20),NAMEI
      CHARACTER*1 A
      CHARACTER*30 PARANAME
      DIMENSION HARA(20),HARB(20),HARC(20),
     +HARD(20),ZC(20),ALQDN(20),TDEN(20),AMW(20),TB(20),TF(20),
     +HVAP(20),Z(20),X2(20),X3(20),X4(20),X5(20),X6(20),X7(20),
     +X8(20),X9(20),X10(20)

C     NCOMP=TOTAL # OF COMPONENTS
C     NLIGHT=# OF COMPONENTS TREATED UN SYMMETRIC CONVENTION.
C     NACTCO=# OF BINARY INTERACTION PARAMETERS PER PAIR.

      WRITE(*,*)' READ THE # OF NLIGHT(NONCONDENSABLE)'
      WRITE(*,*)'     AMONG THE  MIXTURE #',IMIX
      READ(*,*) NLIGHT

      IF (MODEL.EQ.1) NACTCO=2
      IF (MODEL.EQ.5) NACTCO=1
      IF (MODEL.EQ.6) NACTCO=1
      IF (MODEL.EQ.8) NACTCO=2
      IF (MODEL.EQ.9) NACTCO=1
      IF (MODEL.EQ.10)NACTCO=6

      N1=NLIGHT+1


      NCOMP1=NCOMP-1

      M=0
```

```
C
C       READ PURE COMPONENTS PROPERTIES
C
        OPEN(3,FILE='GEN.DAT',STATUS='OLD',ACCESS=
     +'DIRECT',FORM='UNFORMATTED',RECL=431)
7       DO 20 I=1,NCOMP

        READ(3,REC=MIXCOM(IMIX,I))NAMEI,ANTAI,ANTBI,ANTCI,HARAI,
     +HARBI,HARCI,HARDI,TCI,PCI,VCI,ZCI,OMEGAI,ALQDNI,TDENI,
     +AMWI,TBI,TFI,HVAPI,ZRAI,RI,QI,ZI,DIPOI,ETAI,OMGHI,X1I,
     +X2I,X3I,X4I,X5I,X6I,X7I,X8I,X9I,X10I,X11I,X12I,X13I,X14I

        NAME(I)=NAMEI
        ANTA(I)=ANTAI
        ANTB(I)=ANTBI
        ANTC(I)=ANTCI
        HARA(I)=HARAI
        HARB(I)=HARBI
        HARC(I)=HARCI
        HARD(I)=HARDI

        TC(I,I)=TCI
        PC(I,I)=PCI
        VC(I)=VCI
        ZC(I)=ZCI

        OMEGA(I)=OMEGAI
        ALQDN(I)=ALQDNI
        TDEN(I)=TDENI
        AMW(I)=AMWI
        TB(I)=TBI
        TF(I)=TFI
        HVAP(I)=HVAPI
        ZRA(I)=ZRAI
        R(I)=RI
        Q(I)=QI
        Z(I)=ZI
        DIPOLE(I)=DIPOI
        ETA(I)=ETAI
        OMGHI=OMGHI
        QP(I)=X1I
        X2(I)=X2I
        X3(I)=X3I
        X4(I)=X4I
        X5(I)=X5I
        X6(I)=X6I
        X7(I)=X7I
        X8(I)=X8I
        X9(I)=X9I
        X10(I)=X10I
```

```
20      CONTINUE

C       COMPONENTS ARE REARRANGED IN ORDER OF INCREASING CRITICAL
TEMPERATURE
        M=M+1
        IF (M.GT.1.OR.NPROP.EQ.3.OR.NPROP.EQ.4) GO TO 31

        DO 25 J=1,NCOMP1
        DO 23 I=1,NCOMP1

        IF (TC(I,I).GT.TC(I+1,I+1)) THEN
        TE=MIXCOM(IMIX,I)
        TD=TC(I,I)
        MIXCOM(IMIX,I)=MIXCOM(IMIX,I+1)
        TC(I,I)=TC(I+1,I+1)
        MIXCOM(IMIX,I+1)=TE
        TC(I+1,I+1)=TD
        END IF
23      CONTINUE
25      CONTINUE
        GO TO 7
31      DO 30 I=1,NCOMP
        WRITE(*,27)I, NAME(I)
27      FORMAT(1X,'COMPONENT #',I2,' IS',2X,A40)

30      CONTINUE

        DO 90 I=1,NCOMP
        IF(ZRA(I).LE.0.0) THEN
          T1=X5(I)
          T2=X6(I)
          T3=X7(I)
          V1=X8(I)
          V2=X9(I)
          V3=X10(I)
          IF(T3.LE.0.0) THEN
            CVLIQ(I,3)=0.0
              IF(T2.LE.0.0) THEN
                CVLIQ(I,2)=0.0
                CVLIQ(I,1)=V1
              ELSE
                CVLIQ(I,2)=(V2-V1)/(T2-T1)
                CVLIQ(I,1)=V1-CVLIQ(I,2)*T1
              END IF
          ELSE
           CVLIQ(I,3)=((T3-T1)*(V2-V1)-(V3-V1)*(T2-T1))/((T2**2-T1**2)*
     +     (T3-T1)-(T3**2-T1**2)*(T2-T1))
           CVLIQ(I,2)=((V2-V1)-CVLIQ(I,3)*(T2**2-T1**2))/(T2-T1)
           CVLIQ(I,1)=V1-CVLIQ(I,2)*T1-CVLIQ(I,3)*T1**2
          END IF
        END IF
 90     CONTINUE


 C      CALCULATE CRITICAL PROPERTIES FOR BINARY INTERACTIONS
        DO 110 I=1,NCOMP1
        DO 100 J=1,NCOMP

        TC(I,J)=DSQRT(TC(I,I)*TC(J,J))

        PD=VC(I)**0.3333333333333
        QD=VC(J)**0.3333333333333
        RD=(PD+QD)*(PD+QD)*(PD+QD)
        PC(I,J)=4.0*TC(I,J)*(PC(I,I)*VC(I)/TC(I,I)+PC(J,J)*
     +VC(J)/TC(J,J))/RD
        TC(J,I)=TC(I,J)
        PC(J,I)=pC(I,J)
 100    CONTINUE
 110    CONTINUE
 C      FROM THIS PART TO THE END IS TO READ INTERACTION PARAMETER
 C      BY FOLLOWING THE MODEL.
        GO TO (301,302,303,304,305,306,307,308,309,310,311,312,313,
     +  314,315,316,317,318,319,320) MODEL

 301    PARANAME='PARA1.DAT'
        GO TO 115
 302    PARANAME='PARA2.DAT'
        GO TO 115
 303    PARANAME='PARA3.DAT'
        GO TO 115
 304    PARANAME='PARA4.DAT'
        GO TO 115
 305    PARANAME='PARA5.DAT'
        GO TO 115
 306    PARANAME='PARA5.DAT'
        GO TO 115
 307    PARANAME='PARA7.DAT'
        GO TO 115
 308    PARANAME='PARA8.DAT'
        GO TO 115
 309    PARANAME='PARA9.DAT'
        GO TO 115
 310    PARANAME='PARA10.DAT'
        GO TO 115
 311    PARANAME='PARA11.DAT'
        GO TO 115
 312    PARANAME='PARA12.DAT'
        GO TO 115
 313    PARANAME='PARA13.DAT'
        GO TO 115
```

```
314       PARANAME='PARA14.DAT'
          GO TO 115
315       PARANAME='PARA15.DAT'
          GO TO 115
316       PARANAME='PARA16.DAT'
          GO TO 115
317       PARANAME='PARA17.DAT'
          GO TO 115
318       PARANAME='PARA18.DAT'
          GO TO 115
319       PARANAME='PARA19.DAT'
          GO TO 115
320       PARANAME='PARA20.DAT'
          GO TO 115
115       OPEN(2,FILE=PARANAME,STATUS='OLD',ACCESS='DIRECT',FORM=
     +    'UNFORMATTED',RECL=10*NACTCO+1)
          OPEN(3,FILE='THENRYV.DAT',STATUS='OLD',
     +    ACCESS='DIRECT',FORM='FORMATTED',RECL=61)
C         SET ALL INTERACTION PARAMETER ZERO
          DO 119 I=1,NCOMP
          DO 118 J=1,NCOMP
          DO 117 K=1,NACTCO
          CACTCO(I,J,K)=0.0
117       CONTINUE
118       CONTINUE
119       CONTINUE
          IF(NLIGHT)200,200,120
C         READ IN HENRY'S LAW CONSTANTS, PARTIAL MOLAR VOLUMES AT INFINITE
C         DILUTION,AND BINARY INTERACTION PARAMETERS.
120       DO 190 I=1,NLIGHT
          DO 180 J=N1,NCOMP
          WRITE(*,*)' DO YOU WANT INTERACTIVE DATA INPUT'
          WRITE(*,*)' IF YOU WANT PRESS I OR NOT PRESS D'
          READ(*,121)A
121       FORMAT(A)
          IF(A.NE.'I') GO TO 270

          WRITE(*,122)
122       FORMAT(1X,'READ TWO TEMPERATURES T1,T2 + CORRESPONDING
HENRY'/1X,
     + 'CONSTANTS H1,H2 + CORRESPONDING TWO VOLUMS V1,V2')
          READ(*,*)T1,T2,H1,H2,V1,V2
          GO TO 278
270       READ(3,275,REC=(MIXCOM(IMIX,I)-1)*50+MIXCOM(IMIX,J))
     +     (X(K),K=1,6)
          T1=X(1)
          H1=X(2)
          V1=X(3)
          T2=X(4)
          H2=X(5)
```

```
        V2=X(6)

275     FORMAT(1X,6D10.4)
278     READ(2,REC=(MIXCOM(IMIX,I)-1)*50+MIXCOM(IMIX,J))
     +  (X(K),K=1,NACTCO)
        DO 280  K=1,NACTCO
        CACTCO(I,J,K)=X(K)
280     CONTINUE
        DO 285 K=1,NACTCO
        IF(CACTCO(I,J,K).NE.0) THEN
        NREFER(I)=J
        END IF
285     CONTINUE
        IF(V2) 130,130,140
130     CVLIQL(I,J,2)=0.0
        CVLIQL(I,J,1)=V1
        GO TO 150
140     CVLIQL(I,J,2)=(V2-V1)/(T2-T1)
        CVLIQL(I,J,1)=V1-CVLIQL(I,J,2)*T1
150     IF(H2) 160,160,170
160     CHENRY(I,J,2)=0.0
        CHENRY(I,J,1)=H1

        GO TO 180
170     CHENRY(I,J,2)=LOG(H1/H2)/LOG(T1/T2)
        CHENRY(I,J,1)=H1/T1**CHENRY(I,J,2)
180     CONTINUE
190     CONTINUE
C       READ IN BINARY INTERACTION PARAMETERS FOR COMPONENTS TREATED IN
THE
C       SYMMETRIC CONVENTION.
        IF (N1-NCOMP1)200,200,240
200     DO 230 I=N1,NCOMP


        DO 220 J=N1,NCOMP

        READ(2,REC=MIXCOM(IMIX,I)*50-50+MIXCOM(IMIX,J))
     +  (X(K),K=1,NACTCO)
        DO 210 K=1,NACTCO
        CACTCO(I,J,K)=X(K)
210     CONTINUE
220     CONTINUE
230     CONTINUE
        CLOSE(2)

*
* READ ASSOCIATION/SOLVATION PARAMETER
*
        OPEN(2,FILE='ASPARA.DAT',STATUS='OLD',ACCESS='DIRECT',FORM=
     +  'UNFORMATTED',RECL=11)
        DO 235 I=1,NCOMP
          DO 235 J=I,NCOMP

            READ(2,REC=(MIXCOM(IMIX,I)-1)*50+MIXCOM(IMIX,J))X(1)
              ET(I,J)=X(1)
              ET(J,I)=ET(I,J)
              IF(I.EQ.J) ET(I,J)=ETA(I)
235     CONTINUE


240     TOLD=0.0
500     FORMAT(1X,10D10.4)

        RETURN
        END
        SUBROUTINE VIRIAL

*  VIRIAL CALCULATES SECOND VIRIAL COEFFICIENTS,BIJ,FOR ALL PAIRS OF N
* COMPONENTS (N.LE.20). IF CARBOXYLIC ACIDS ARE PRESENT KV=1(OTHERWISE
* 0).
        IMPLICIT REAL*8(A-H,O-Z)
        INTEGER GOAL
        COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     +  SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     +  OMEGA(20),F(20)
        COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX,
     +  MIXCOM(50,10),NCM(10)
        COMMON/VIRIAL/B(20,20),BD(20,20),BF(20,20),DB(20,20),
     +  DBD(20,20),ETA(20),DIPOLE(20),ET(20,20)
        CHARACTER*40 NAME(20)
        DIMENSION G(20,20),TS(20,20),
     +S(20,20),Z(20,20),H(20,20),W(20,20),E(20,20),A(20,20)

        DATA B1,B2,B3/1.2618,7243.8,1.7941E07/,CN1,CN2,CN3,CN4/0.94,
     1  -1.47,-.85,1.015/,CP1,CP2,CP3,CP4/-.75,3.,-2.1,-2.1/,CA1,
     2  CA2,CH1,CH2/-0.3,-0.05,1.99,0.2/,CW1,CW2,CW3/0.006026,0.02096,
     3  -0.001366/,CS1,CK1,CK2,CE1,CE2,CE3,CE4,CE5/2.4507,0.7,0.6,650.,
     4  300.,4.27,42800.,22400./,CD1,CD2,CD3/0.748,0.91,0.4/,
     5  E3/0.3333333/

* CALCULATE TEMPERATURE-INDEPENDENT PARAMETERS ONLY FOR NEW SYSTEM
* RESET ASSOCIATING VAPOR FLAG


109     KV=0
* CALCULATE TEMPERATURE-INDEPENDENT PARAMETERS FOR PURE COMPONENTS
110     DO 119 I=1,NCOMP
* MOLECULAR SIZE PARAMETER (CUBED)

         W(I,I)=OMEGA(I)


         S(I,I)=(CS1-W(I,I))**3*(1.013*TC(I,I)/PC(I,I))


         IF(S(I,I).LT.0.) GO TO 900
         IF(ETA(I).GE.4.4999) KV=1

* ENERGY PARAMETER
       TS(I,I)=TC(I,I)*(CD1+CD2*W(I,I)-CD3*ETA(I)/(2.+20.*W(I,I)))

         IF(TS(I,I).LT.0.) GO TO 900
         IF(DIPOLE(I).LT.1.45) GO TO 117
```

```
* MODIFICATION OF PARAMETERS FOR LARGE DIPOLE MOMENTS
          H1=16.+400.*W(I,I)
          T1=H1/(H1-6.0)
          T2=3./(H1-6.0)
          TK=2.882-(1.882*W(I,I)/(0.03+W(I,I)))
          Z(I,I)=B3*DIPOLE(I)**4/(TS(I,I)*S(I,I)**2*TC(I,I)*TK)
          IF (Z(I,I).LT.-1.) GO TO 900
* MODIFIED MOLECULAR SIZE PARAMETER(CUBED)
          S(I,I)=S(I,I)*(1.+T2*Z(I,I))
* MODIFIED ENERGY PARAMETER
          TS(I,I)=TS(I,I)*(1.-T1*Z(I,I)+T1*(T1+1.)*Z(I,I)**2/2.)
* REDUCED DIPOLE MOMENT
117       G(I,I)=B2*DIPOLE(I)**2/(TS(I,I)*S(I,I))
119     CONTINUE
        IF(NCOMP.EQ.1) GO TO 130
* CALCULATE TEMPERATURE-INDEPENDENT PARAMETERS FOR COMPONENT PAIRS
120     DO 129 I=2,NCOMP
        DO 129 J=1,I-1
* CROSS NONPOLAR ACENTRIC PARAMETER
          W(I,J)=(W(I,I)+W(J,J))/2.
* CROSS MOLECULAR SIZE PARAMETER
          S(I,J)=SQRT(S(I,I)*S(J,J))
* CROSS ENERGY PARAMETER
        TS(I,J)=CK1*SQRT(TS(I,I)*TS(J,J))+CK2/(1./TS(I,I)+1./TS(J,J))
          IF (DIPOLE(I).GE.2.0.AND.DIPOLE(J).LT.1.E-19) THEN
            Z(I,J)=DIPOLE(I)**2*(TS(J,J)**(2./3.)*S(J,J)**(4./3.))/
     1      (TS(I,J)*S(I,J)**2)
            GO TO 125
          ELSE
            IF (DIPOLE(J).GE.2.0.AND.DIPOLE(I).LT.1.E-19) THEN

Z(I,J)=DIPOLE(J)**2*(TS(I,I)**(2./3.)*S(I,I)**(4./3.))/(TS(I,J)
     2      *S(I,J)**2)
            GO TO 125
            ELSE
            END IF
          Z(I,J)=0
          GO TO 126
        END IF
125     H1=16.+400.*W(I,J)
        T1=H1/(H1-6.)
        T2=3./(H1-6.)
* MODIFIED CROSS MOLECULAR SIZE PARAMETER
        S(I,J)=S(I,J)*(1.-T2*Z(I,J))
* MODIFIED CROSS ENERGY PARAMETER
        TS(I,J)=TS(I,J)*(1.+T1*Z(I,J))
* CROSS REDUCED DIPOLE MOMENT
126     G(I,J)=B2*DIPOLE(I)*DIPOLE(J)/(TS(I,J)*S(I,J))
* MAKE DESCRIMINATION OF ASSOCIATING SYSTEM
        IF(ET(I,J).GE.4.4999) KV=1
```

```
129     CONTINUE
* CALCULATE TEMPERATURE-INDEPENDENT TERMS IN VIRIAL COEFFICIENTS FOR
* PURE COMPONENTS AND PAIRS.
130     DO 139 I=1,NCOMP
        DO 139 J=1,I
        S(I,J)=B1*S(I,J)
        H(I,J)=CH1+CH2*G(I,J)**2
        A(I,J)=CA1+CA2*G(I,J)
* DETERMINE MODIFIED REDUCED DIPOLE PARAMETER
        IF (G(I,J).LT.0.04) GO TO 135
        IF (G(I,J).GE.0.25) GO TO 134
        G(I,J)=0.
        GO TO 135
134     G(I,J)=G(I,J)-0.25
135     IF (ET(I,J).GE.4.49999) GO TO 137
        IF (ET(I,J).LT.1.E-19) GO TO 139
* ENERGY TERM FOR NONASSOCIATING PAIR
        E(I,J)=CE1/(TS(I,J)+CE2)-CE3
        GO TO 139
* ENERGY TERM FOR ASSOCIATING PAIR
137     E(I,J)=CE4/(TS(I,J)+CE5)-CE3
139     CONTINUE

* CALCULATE TEMPERATURE-DEPENDENT TERMS AND VIRIAL COEFFICIENTS
200     DO 209 I=1,NCOMP
        DO 209 J=1,I
        TA=T/TS(I,J)
        T1=1./(1./TA-1.6*W(I,J))
        T2=T1*T1
        T3=T2*T1
* NONPOLAR FREE CONTRIBUTION
        BN=CN1+CN2/T1+CN3/T2+CN4/T3
        IF(G(I,J).GT.1.E-19) GO TO 201
        BP=0.
        GO TO 202
* POLAR FREE CONTRIBUTION
201     BP=(CP1+CP2/T1+CP3/T2+CP4/T3)*G(I,J)
* TOTAL FREE CONTRIBUTION TO VIRIAL COEFFICIENT
202     BF(I,J)=S(I,J)*(BN+BP)
        BF(J,I)=BF(I,J)
* METASTABLE PLUS BOUND CONTRIBUTIONS
        BMB=A(I,J)*EXP(H(I,J)/TA)
        IF(ET(I,J).LT.1.E-19) GO TO 204
* CHEMICAL CONTRIBUTION
        BC=EXP(ET(I,J)*E(I,J))-EXP(ET(I,J)*(1500./T+E(I,J)))
        GO TO 205
204     BC=0.
* METASTABLE,BOUND,AND CHEMICAL CONTRIBUTIONS TO VIRIAL COEFFICIENT
205     BD(I,J)=S(I,J)*(BMB+BC)
```

```
208     IF(KV.EQ.0) B(I,J)=BF(I,J)+BD(I,J)
        B(J,I)=B(I,J)
209     CONTINUE
        RETURN
* ERROR RETURN FOR FAILURE TO FIND VALID VIRIAL COEFFICIENTS.
900     DO 902 I=1,NCOMP
        DO 902 J=1,NCOMP
        B(I,J)=0.
902     BD(I,J)=0.
        WRITE(*,*)'WE GET WRONG RESULTS IN 2ND VIRIAL COEFFICIENTS'
        RETURN
        END
```

```
      SUBROUTINE PHIMIX
* PHIMIX CALCULATES VAPOR PHASE FUGACITY COEFFICIENTS,PHI,FOR ALL N
* COMPONENTS (N.LE.20) WHOSE INDICES APPEAR IN VECTOR ID,GIVEN
* TEMPERATURE T(K),PRESSURE P,AND VAPOR COMPOSITION Y, THIS PROGRAM
* CONTAINS CHEMICAL THEORY TO CALCULATE VAPOR PHASE, WHICH CONTAIN
* CARBOXYLIC ACID ,FUGACITY.

      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     + SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     + OMEGA(20),F(20)
      COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX,
     + MIXCOM(50,10),NCM(10)
      COMMON/VIRIAL/B(20,20),BD(20,20),BF(20,20),DB(20,20),
     + DBD(20,20),ETA(20),DIPOLE(20),ET(20,20)
      COMMON/PHIS/PHI(20),PHL(20),BMIX
      CHARACTER*40 NAME(20)

      DIMENSION SI(20),ZO(20),SS(20),RM(20),C(20,20),ZI(20)
      DATA R/82.057/

      PRT=P/(R*T)

* CHECK FOR SIGNIFICANT CHANGE IN T OR P SINCE LAST CALL FOR SYSTEM
      IF(ABS(T-TOLD).LT.0.02.AND.ABS(P-POLD).LT.0.01) THEN
        IF (KV.EQ.1) GO TO  200

* GET SECOND VIRIAL COEFFICIENTS BIJ IN /VIRIAL/
      ELSE
        CALL VIRIAL
        IF(KV.EQ.1) GO TO 200
      END IF
* CALCULATE SECOND VIRIAL COEFFICIENT FOR GAS MIXTURE,BMIX
      BMIX=0.
      DO 139 I=1,NCOMP
* CALCULATE EFF SECOND VIRIAL COEFFICIENT FOR COMP I IN MIXTURE,SS(I)

        SS(I)=0.
          DO 133 J=1,NCOMP
133         SS(I)=SS(I)+Y(J)*B(I,J)

        BMIX=BMIX+Y(I)*SS(I)

* NORMALIZATION OF SS(I),BMIX, INCASE OF UNKNOWN VAPOR COMPOSITION Y
        SS(I)=SS(I)/SUMY
139   CONTINUE
      BMIX=BMIX/SUMY**2
```

```
* CALCULATE VAPOR PHASE FUGACITY COEFFICIENTS,PHI(I)
140     DO 149 I=1,NCOMP
          PHI(I)=DEXP(PRT*(2.*SS(I)-BMIX))
149       PHL(I)=PHI(I)


* SAVE CONDITIONS AT WHICH PHIS CALCULATED

        RETURN
* SPECIAL CALCULATION FOR ASSOCIATING GAS MIXTURES

* IF PREVIOUS PHI VALUES AVAILABLE USE TO GET FIRST ESTIMATES OF ACTUAL
* VAPOR COMPOSITION


* FOR NO PREVIOUS PHI VALUES AVAILABLE  MAKE FIRST ESTIMATES OF
* ACTUAL VAPOR COMPOSITION
* FOR ALL CASES , FIND VALUES OF ASSOCIATION EQUILIBRIUM VALUES

200     DO 207 I=1,NCOMP
          DO 207 J=1,I

            CM=-2.*PRT*BD(I,J)*DEXP(PRT*(BF(I,I)+BF(J,J)-BF(I,J)))
            IF (CM.LT.0) GO TO 900
            IF(I.EQ.J) THEN
              C(I,J)=CM/2.0

                IF (C(I,J).LE.0.5) THEN
                  ZI(I)=Y(I)
                ELSE
                  ZT=(SQRT(1.+8.*C(I,J)*Y(I))-1.)/(4.*C(I,J))
                  IF(ZT.LT.ZI(I)) ZI(I)=ZT
                END IF
            ELSE
              C(I,J)=CM
                IF (C(I,J).GT.0.5) THEN
                  IF (Y(J).GT.Y(I)) THEN
                    ZT=Y(I)/(C(I,J)*Y(J)+1.)
                    ZJ=Y(J)/(C(I,J)*ZI(I)+1.)
                    IF(ZT.LT.ZI(I)) ZI(I)=ZT
                    IF(ZJ.LT.ZI(I)) ZI(J)=ZJ
                  ELSE

                    ZJ=Y(J)/(C(I,J)*Y(I)+1.)
                    ZT=Y(I)/(C(I,J)*ZI(J)+1.0)
                    IF(ZT.LT.ZI(I)) ZI(I)=ZT
                    IF(ZJ.LT.ZI(I)) ZI(I)=ZJ
                  END IF
                ELSE
                  ZI(I)=Y(I)
                END IF
              END IF
207       CONTINUE
* START ITERATIVE CALCULATION OF ACTUAL VAPOR COMPOSITION,ZI(I)
* STORE FIRST ITERATION VALUES

        DO 209 I=1,NCOMP
209       ZO(I)=ZI(I)

        IT=0
210     IT=IT+1
        IF(IT.GT.20) THEN
          ZI(I)=PHL(I)*Y(I)*DEXP(-PRT*BF(I,I))
          ZO(I)=ZI(I)
          IT=0
        ELSE
        END IF
        RS=0.
220     DO 229 I=1,NCOMP
          SI(I)=0.
* DAMP ITERATION 20 PERCENT
          ZI(I)=.2*ZO(I)+0.8*ZI(I)
          ZO(I)=ZI(I)
            DO 221 J=1,I
221           SI(I)=SI(I)+C(I,J)*ZI(J)
            DO 223 J=I,NCOMP
223           SI(I)=SI(I)+C(J,I)*ZI(J)

          RM(I)=ZI(I)*SI(I)
          RS=RS+RM(I)
229     CONTINUE
230     DO 235 I=1,NCOMP
235       ZI(I)=(1.+(1./2.)*RS)*Y(I)/(1.+SI(I))
        DO 239 I=1,NCOMP
          IF (Y(I).LT.1.E-09) GO TO 239
* CHECK CONVERGENCE OF EACH ZI(I)
          IF (ABS((ZI(I)-ZO(I))/Y(I)).GT.0.005) GO TO 210
239     CONTINUE
* CALCULATE VAPOR PHASE FUGACITY COEFFICIENTS FOR ACTUAL COMPOSITION OF
* ASSOCIATING VAPOR.
240     DO 249 I=1,NCOMP
          PHI(I)=((1+0.5*RS)/(1.+SI(I)))*DEXP(PRT*BF(I,I))
* SAVE FUGACITY COEFFICIENTS FOR USE AT SIMULAR CONDITIONS
          PHL(I)=PHI(I)
249     CONTINUE
        RETURN
* ERROR RETURN FOR FAILURE OF ITERATION FOR ZI(I) TO CONVERGE.
900     DO 901 I=1,NCOMP
          PHI(I)=1
901       PHL(I)=1
```

```
WRITE(*,*)'YOU GOT THE WRONG PHI VALUE'
RETURN
END
```

```
      SUBROUTINE RSTATE
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     + SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     + OMEGA(20),F(20)
      COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX,
     + MIXCOM(50,10),NCM(10)
      COMMON/RST/HENRY(10,30),CHENRY(10,30,2),FREFER(20),
     + ZRA(20),VLIQ(20),VLIQL(20,20),CVLIQL(20,20,2),CVLIQ(20,3)
      COMMON/VIRIAL/B(20,20),BD(20,20),BF(20,20),DB(20,20),
     + DBD(20,20),ETA(20),DIPOLE(20),ET(20,20)
      COMMON/PSAT/ANTA(20),ANTB(20),ANTC(20),PSAT(20)
      CHARACTER*40 NAME(20)


     FSAT0F(TR)=(-1.1970522/TR-1.3785023)/TR+2.0048841
     FSAT1F(TR)=(-2.7741817/TR+1.5454928)/TR+1.3057555
    PHIS0F(TR)=((.57335015/TR-3.076574)/TR+5.6085595)/TR-3.502135
   +8
    PHIS1F(TR)=(((((((((((.012089114/TR-.015172164)/TR-.068603516
   +)/TR+.024364816)/TR+0.14936909)/TR+0.18927037)/TR-0.12147436)/TR
   +-0.10665730)/TR-1.1662283)/TR+0.12666184)/TR+0.3166137)/TR+
   +4.3538729)/TR-3.7694018


      N1=NLIGHT+1
      IF((T-TOLD).EQ.0) GO TO 381
      RT=82.057*T
      NCRIT=NLIGHT

      DO 100 J=1,NCOMP
      TR=T/TC(J,J)
      IF(TR.GT.0.75) THEN
        TAU=1.60+0.00693026/(TR-0.655)
      ELSE
        TAU=1.+(1.-TR)**0.28571429
      END IF

      IF(ZRA(J).LE.0.0) THEN
        VLIQ(J)=CVLIQ(J,1)+CVLIQ(J,2)*T+CVLIQ(J,3)*T**2
      ELSE
        VLIQ(J)=82.057*TC(J,J)*ZRA(J)**TAU/PC(J,J)
      END IF
      PSAT(J)=(EXP(ANTA(J)-ANTB(J)/(T+ANTC(J))))/760

100   CONTINUE

      DO 370 J=N1,NCOMP
      W=OMEGA(J)
```

```
        IF((TR-1.0).GT.0) THEN
         NCRIT=J
         GO TO 370
        END IF


        IF ((TR-.56).GT.0) THEN
        PHIS=EXP(PHIS0F(TR)+W*PHIS1F(TR))
        ELSE
        PHIS=EXP(B(J,J)*PSAT(J)/RT)
        END IF
        FREFER(J)=PSAT(J)*PHIS*EXP(-VLIQ(J)*PSAT(J)/RT)
370     CONTINUE

381     IF ((NCRIT).LE.0) GO TO 430

        IF((NCRIT-NLIGHT).EQ.0) GO TO 400

        DO 390 J=N1,NCRIT

        FREFER(J)=PC(J,J)*EXP(FSAT0F(TR)+OMEGA(J)*FSAT1F(TR))


390     CONTINUE
        IF (NLIGHT.LE.0) GO TO 430
400     NC1=NCRIT+1
        VOLSUM=0.0
        DO 405 J=NC1,NCOMP
405     VOLSUM=VOLSUM+X(J)*VLIQ(J)

        DO 420 J=1,NLIGHT
        VOL=0.0
        DO 410 I=NC1,NCOMP
        VLIQL(J,I)=CVLIQL(J,I,1)+CVLIQL(J,I,2)*T
        HENRY(J,I)=CHENRY(J,I,1)*T**CHENRY(J,I,2)*EXP(-VLIQL(J,I)*PSAT
     +(I)/RT)
        VOL=V0L+X(I)*VLIQ(I)*VLIQL(J,I)


410     CONTINUE
        NR=NREFER(J)
        FREFER(J)=HENRY(J,NR)
        VLIQ(J)=VOL/VOLSUM
420     CONTINUE

430     RETURN
        END
```

```
          SUBROUTINE ACTCO

          COMMON/CNTRL/MODEL,GOAL,NPROP,NC,NG

          GO TO (115,116,117,118,119,120,121,122,123,124,125,126,127,
     +    128,129,130,131,132,133,134) MODEL
 115      CALL ACTCO1
          GO TO 145
 116      CALL ACTCO2
          GO TO 145
 117      CALL ACTCO3
          GO TO 145
 118      CALL ACTCO4
          GO TO 145
 119      CALL ACTCO5
          GO TO 145
 120      CALL ACTCO5
          GO TO 145
 121      CALL ACTCO7
          GO TO 145
 122      CALL ACTCO8
          GO TO 145
 123      CALL ACTCO9
          GO TO 145
 124      CALL ACTCO10
          GO TO 145
 125      CALL ACTCO11
          GO TO 145
 126      CALL ACTCO12
          GO TO 145
 127      CALL ACTCO13
          GO TO 145
 128      CALL ACTCO14
          GO TO 145
 129      CALL ACTCO15
          GO TO 145
 130      CALL ACTCO16
          GO TO 145
 131      CALL ACTCO17
          GO TO 145
 132      CALL ACTCO18
          GO TO 145
 133      CALL ACTCO19
          GO TO 145
 134      CALL ACTCO20
          GO TO 145

 145      RETURN
          END
```

```
      SUBROUTINE ACTCO1

      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER GOAL

      COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     + SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     + OMEGA(20),F(20)
      COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX,
     + MIXCOM(50,10),NCM(10)
      COMMON/CNTRL/MODEL,GOAL,NPROP,NC,NG
      COMMON/RST/HENRY(10,30),CHENRY(10,30,2),FREFER(20),
     + ZRA(20),VLIQ(20),VLIQL(20,20),CVLIQL(20,20,2)
      COMMON/ACTCO/CACTCO(20,20,6),GAMMA(20),HERT
      COMMON /C/ C


      CHARACTER*40 NAME(20)
      CHARACTER*1 A

      DIMENSION C(20,20),ZI(20),PHI(20),YI(20)
      DIMENSION E(20,20),ZIJ(20,20),XIJ(20,20)
      DIMENSION IE(20,20),IZIJ(20,20),ELNGM(20)

C
C       First time through we obtain all constants a .d initial
C           parameter values
C

      DO 5 I=1,NCOMP
        DO 5 J=1,NCOMP
          ZIJ(I,J)=CACTCO(I,J,1)
          E(I,J)=CACTCO(I,J,2)
5     CONTINUE

      DO 7 I=1,NCOMP
        DO 7 J=1,NCOMP

           IZIJ(I,J)=0.0
           IF(ZIJ(I,J) .LT. 0.0D0) IZIJ(I,J)=1
           IF(ZIJ(I,J) .LT. -1.5D0 .AND. I.NE.J) IZIJ(I,J)=-1
 7    CONTINUE

      DO 8 I=1,NCOMP
        DO 8 J=I,NCOMP
          E(J,I)=E(I,J)

          IE(I,J)=0.0
```

```
            IF(E(I,J) .LT. -199.0D0) IE(I,J)=1
            IE(J,I)=IE(I,J)

 8      CONTINUE

* IF DATA POINT NUMBER IS 1,THEN DETERMINE THE NEW SET OF VALUES
* FOR THE ADJUSTIABLE PARAMETERS

   22       DO 16 I=1,NCOMP
               DO 16 J=I,NCOMP

* E(I,J) VALUE MUST BE FOUND BY REGRESSION IN CASE OF POOR DATA
                  IF (IE(I,J) .EQ. 1) THEN
                    E(I,J)=0.0
                    E(J,I)=E(I,J)
                  END IF
   16           CONTINUE
            DO 161 I=1,NCOMP-1
               C(I,I)=1.0D0
               I1=I+1
               DO 161 J=I1,NCOMP
                  C(I,J)=DEXP(E(I,I)+E(J,J)-2.0D0*E(I,J))
                  C(J,I)=C(I,J)
  161             CONTINUE
            C(NCOMP,NCOMP)=1.0D0

            DO 17 I=1,NCOMP

* ZIJ(I,J) VALUE MUST BE FOUND BY REGRESSION IN CASE OF POOR DATA
               IF (IZIJ(I,I) .EQ. 1) THEN
                  ZIJ(I,I)=0.0
               END IF

   18          DO 17 J=1,NCOMP
                  IF (J .NE. I) THEN
                    IF (IZIJ(I,J) .LT. 0) ZIJ(I,J)=ZIJ(I,I)
* ZIJ(I,J) VALUE MUST BE FOUND BY REGRESSION IN CASE OF POOR DATA
                    IF (IZIJ(I,J) .EQ. 1) THEN
                       ZIJ(I,J)=0.0
                    END IF

                  END IF
   17          CONTINUE


C
C        Calculation by 3-D Lattice Gas Analogy
C            or Guggenheim Analogy
C
```

```
   42 DO 32 I=1,NCOMP
        ZI(I)=0.0D0
   32   CONTINUE
     SUM=0.0D0
     SUMV=0.0D0
     DO 33 I=1,NCOMP
       DO 34 J=1,NCOMP
         ZI(I)=ZI(I)+ZIJ(I,J)*X(J)
   34    CONTINUE
       SUM=SUM+ZI(I)*X(I)
       SUMV=SUMV+VLIQ(I)*X(I)
   33  CONTINUE

     DO 35 I=1,NCOMP
       YI(I)=ZI(I)*X(I)/SUM
       IF (SUMV.NE. 0.0D0) PHI(I)=VLIQ(I)*X(I)/SUMV

   35  CONTINUE
     CALL XSOLVE(NCOMP,YI,XIJ)


  37   DO 38 I=1,NCOMP
         ELNGM(I)=0.0D0
         IF (X(I) .EQ. 0.0D0) GO TO 38

           ELNGM(I)=DLOG(PHI(I)/X(I))+ZI(I)*
    +        DLOG(XIJ(I,I)/PHI(I))/2.0D0+(1.0D0-PHI(I)/X(I))-
    +        ZI(I)*(1.0D0-PHI(I)/YI(I))/2.0D0
    +        +(ZI(I)-ZIJ(I,I))*E(I,I)/2.0D0
          DO 39 J=1,NCOMP
             IF (X(J) .EQ. 0.0D0) GO TO 39
              ELNGM(I)=ELNGM(I)+X(J)*(ZIJ(J,I)
    +            -ZI(J))*(E(J,J)+DLOG(XIJ(J,J)/PHI(J)))/2.0D0
  39          CONTINUE

  38    CONTINUE


     DO 50 I=1,NCOMP
       GAMMA(I)=DEXP(ELNGM(I))
50   CONTINUE


      HERT=0.0D0
      DO 36 I=1,NCOMP-1
        I1=I+1
        DO 36 J=I1,NCOMP
          HERT=HERT-ZI(I)*X(I)*XIJ(I,J)*DLOG(C(I,J))/2.0D0
  36      CONTINUE
     write(*,*)'hert1',hert
```

```
         DO 361 I=1,NCOMP
            HERT=HERT+(ZI(I)-ZIJ(I,I))*X(I)*E(I,I)/2.0D0
        write(*,*)i,e(i,i)
  361       CONTINUE
        write(*,*)'hert2',hert
C
        RETURN
      END
      SUBROUTINE XSOLVE(NCOMP,YI,XIJR)
C
C         This subroutine employs a fixed point iteration
C           method to solve for the R ratios, and then
C           calculates the local compositions Xij that
C           R-values represent.
C
      IMPLICIT REAL*8(A-H,O-Z)
      LOGICAL TEST
      DIMENSION YI(20),XIJR(20,20)
      DIMENSION C(20,20),CSQRT(20,20)
      DIMENSION R(20),RNEW(20)
      COMMON /C/ C
      DATA EPS/1.0D-06/
        IT=-1
        K=1
C
C      Determine key component by largest composition value
C
        DO 1 I=1,NCOMP
         IF(YI(I).GT.YI(K)) K=I
   1     CONTINUE
        SUM=0.0D0
C
C      Calculation of initial guesses.  We do so by ignoring
C         the presence of all components other than the key
C         component and the component for which we are
C         obtaining an initial guess.
C      We also calculate the CSQRT values here to save needless
C         repitition in the iteration loop.
C
      DO 5 I=1,NCOMP
      IF(I.EQ.K) THEN
         R(I)=1.0D0
         SUM=SUM+R(I)

      ELSE
        IF(YI(I).EQ.0.0D0) THEN
          R(I)=0.0D0
          SUM=SUM+R(I)
        ELSE
          T1=YI(I)/YI(K)
          T2=1.0D0+T1
          T3=(C(I,K)-1.0D00)/C(I,K)
          R(I)=2.0D0*T1/(1.0D0-T1+DSQRT(T2*T2-4.0D0*T1*T3))
          SUM=SUM+R(I)
        END IF
      END IF
         DO 5 J=1,NCOMP
            CSQRT(I,J)=DSQRT(C(I,J)/(C(I,K)*C(J,K)))
    5    CONTINUE
            IT=0
C
C      Enter iterative loop
C
    4            DO 9 I=1,NCOMP
                   SUMJ=0.0D0
                   DO 8 J=1,NCOMP
                     SUMJ=SUMJ+R(J)*CSQRT(I,J)
    8                CONTINUE
                   RNEW(I)=(YI(I)/YI(K))*SUM/SUMJ
                   IF(I.EQ.K) RNEW(I)=1.0D0
    9            CONTINUE
                 TEST=.TRUE.
                 DO 10 I=1,NCOMP
                   IF(DABS(RNEW(I)-R(I)).GT.EPS) TEST=.FALSE.
   10              CONTINUE

                 IF(TEST) GO TO 50
   11            DO 12 I=1,NCOMP
                   R(I)=RNEW(I)
   12            CONTINUE
                 IT=IT+1
                 IF(IT.GT.40) GO TO 50
                 SUM=0.0D0
                 DO 7 L=1,NCOMP
                   SUM=SUM+R(L)
    7              CONTINUE
                 GO TO 4
C
C     Have exited loop;  we now calculate the Xij values
C
   50 DO 51 I=1,NCOMP
         SUMJ=0.0D0
         DO 52 J=1,NCOMP
            SUMJ=SUMJ+R(J)*CSQRT(I,J)
   52       CONTINUE
         DO 51 J=1,NCOMP
            XIJR(I,J)=R(J)*CSQRT(I,J)/SUMJ
   51       CONTINUE
      IF(IT.GT.40) WRITE(*,53)
   53 FORMAT('-Maximum number of iterations exceeded')
   DO 54 I=1,NCOMP
   DO 54 J=1,NCOMP
   IF (XIJR(I,J) .LT. 0.0D0) THEN
     WRITE(*,56) (R(K),K=1,NCOMP)
56   FORMAT('- R-VALUES ARE:',3X,8E13.6)
     WRITE(*,57) (YI(K),K=1,NCOMP)
57   FORMAT('- YI-VALUES ARE:',3X,8E13.6/'- C-VALUES ARE:')
     DO 58 L=1,NCOMP
     WRITE(*,59) (C(L,M),M=1,NCOMP)
59   FORMAT('0',8E13.6)
58   CONTINUE
     RETURN

   END IF
54 CONTINUE
   RETURN
   END
```

```
      SUBROUTINE ACTCO5
* ACTCO5 CALCULATES LIQUID PHASE ACTIVITY COEFFICIENTS, GAM, AT GIVEN
* TEMPERATURE T(K) AND LIQUID COMPOSITION X,USING THE UNIQUAC MODEL OR
* MODIFIED UNIQUAC MODEL.
* FOR NONCONDENSABLE COMPONENTS (U(I,I) SET TO 1.E+20) AND UNSYMMETRIC
* CONVENTION IS USED TO DERIVE EFFECTIVE ACTIVITY COEFFICIENTS.
      IMPLICIT REAL*8(A-H,O-Z)
      INTEGER GOAL

      COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     + SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     + OMEGA(20),F(20)
      COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX,
     + MIXCOM(50,10),NCM(10)
      COMMON/CNTRL/MODEL,GOAL,NPROP,NC,NG

      COMMON/PHIS/PHI(20),PHL(20),BMIX
      COMMON/ACTCO/CACTCO(20,20,6),GAMMA(20),HERT
      COMMON/ACTCOU/R(20),Q(20),QP(20)
      CHARACTER*40 NAME(20)
      DIMENSION PT(20),PTS(20),RL(20),TH(20),TP(20),GCL(20),TAU(20,20)
     + ,RU(20),QU(20),U(20,20),TS(20)
      DATA Z/10./
* PARAMETER SUBSTITUTION TO PREPARE  FOR THE PARAMETER MODIFICATION
      BU=0.0

      DO 50 I=1,NCOMP
      DO 50 J=1,NCOMP
        U(I,J)=CACTCO(I,J,1)
        RU(I)=R(I)
        IF(MODEL.EQ.6) THEN
          QP(I)=QP(I)
        ELSE
          QP(I)=Q(I)
        END IF
        QU(I)=Q(I)

50    CONTINUE

* CALCULATE COMPOSITION INDEPENDENT TERMS
      DO 119 I=1,NCOMP
119     RL(I)=Z*(RU(I)-QU(I))/2.-RU(I)+1.
* CALCULATE SEGMENT AND AREA FRACTIONS FOR COMPONENTS IN MIXTURE
      SP=1.E-30
      ST=1.E-30
      STP=1.E-30
      SS=0.
      SL=0.
      DO 125 I=1,NCOMP
```

```
          TH(I)=X(I)*QU(I)
          TP(I)=X(I)*QP(I)
          SP=SP+X(I)*RU(I)
          ST=ST+TH(I)
          STP=STP+TP(I)
* SKIP FOR NONCONDENSABLE COMPONENTS
          IF(U(I,I).LE.1.E+19) THEN
            SS=SS+X(I)
            SL=SL+X(I)*RL(I)

          END IF
125     CONTINUE
        DO 129 I=1,NCOMP
          TH(I)=TH(I)/ST
          TP(I)=TP(I)/STP

          IF(U(I,I).LE.1.E+19) THEN
* CALCULATE COMBINATORIAL CONTRIBUTION TO EXCESS FREE ENERGY
            GCL(I)=RL(I)-RU(I)*SL/SP+DLOG(RU(I)*SS/SP)+(Z/2.)*QU(I)
     +      *DLOG(QU(I)*SP/(RU(I)*ST))
          ELSE
            GCL(I)=0.
          END IF
129     CONTINUE
* GET UNIQUAC BINARY INTERACTION PARAMETER TERMS
* TAUS CALCULATES TEMPERATURE DEPENDENT INTERACTION COEFFICIENTS TAU FOR
* USE IN SUBROUTINE GAMMA. IF SYSTEM DATA ARE MISSING (SOME REQUIRED
* ENTRY IN MATRIX U IN COMMON/BINARY IS ZERO) CORRESPONDING TAU IS
* SET TO 1.
        DO 130 I=1,NCOMP
* CHECK IF J EQUALS I.
* CHECK IF BINARY PAIR ARE BOTH NONCONDENSABLES.
* CHECK IF EITHER COMPONENT IN BINARY PAIR IS A NONCONDENSABLE.
* CHECK IF BINARY DATA ARE MISSING.
        DO 130  J=1,NCOMP
          IF ((J.EQ.I).OR.(U(I,I).GT.1.E+19.AND.U(J,J).GT.1.E+19).OR.
     +    (U(I,I).GT.1.E+19.OR.U(J,J).GT.1.E+19).OR.(ABS(U(I,J)).LT.1.
     +    E-19))   THEN
            TAU(I,J)=1.
          ELSE
* CALCULATE INTERACTION TERM
            TAU(I,J)=EXP(-U(I,J)/T)
          END IF
130     CONTINUE
* CALCULATE RESIDUAL CONTRIBUTION TO EXCESS FREE ENERGY
        DO 141 I=1,NCOMP
141       PTS(I)=0.

        DO 149 I=1,NCOMP
          PT(I)=1.E-30
```

```
              DO 143 J=1,NCOMP
 143            PT(I)=PT(I)+TP(J)*TAU(J,I)
              DO 145 J=1,NCOMP
 145            PTS(J)=PTS(J)+TP(I)*TAU(J,I)/PT(I)

 149        CONTINUE

 150        DO 159 I=1,NCOMP
              IF (U(I,I).LE.1.E+19) THEN
* RESIDUAL FREE ENERGY FOR CONDENSABLE COMPONENTS
                GRL=QP(I)*(1.-DLOG(PT(I))-PTS(I))
              ELSE
                GRL=0.
                DO 156 J=1,NCOMP
* RESIDUAL FREE ENERGY FOR NONCONDENSABLE COMPONENTS
 156              GRL=GRL+TH(J)*(U(I,J)+U(J,I)/T)


              END IF
* CALCULATE  ACTIVITY COEFFICIENT

 158          GAMMA(I)=EXP(GCL(I)+GRL)
 159        CONTINUE

* CALCULATION OF EXCESS ENTHALPY CONTRIBUTION FOR LIQUID
            HE=0.
            HC=0.

 161        DO 169 I=1,NCOMP
            TM1=0.
            TM2=0.
            IF(U(I,I).GT.1.E+19) THEN
            DO 163 J=1,NCOMP
 163        TM1=TM1+TH(J)*U(J,I)
            HC=HC+X(I)*TM1

            ELSE

            DO 165 J=1,NCOMP
            TM3=TP(J)*TAU(J,I)
            TM1=TM1+TM3
 165        TM2=TM2+TM3*(U(J,I)-2.*BU/T)
            HE=HE+QP(I)*X(I)*TM2/TM1
            END IF

 169        CONTINUE
            HERT=(HE+HC)/T
            RETURN
            END
```

```
      SUBROUTINE ACTCO8
C     SUBROUTINE TO GET A ACTIVITY COEFFICIENT BY NRTL MODEL
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     + SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     + OMEGA(20),F(20)
      COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX,
     + MIXCOM(50,10),NCM(10)
      COMMON/ACTCO/CACTCO(20,20,6),GAMMA(20),HERT
      CHARACTER*40 NAME(20)
      DIMENSION ALPHA(30,30),GGIJ(30,30),GIJ(30,30),XIJ(30,30),
     + R(30)
      RT=1.987*T
      DO 10 I=1,NCOMP
      DO 10 J=1,NCOMP
      GIJ(I,J)=CACTCO(I,J,1)/RT
      ALPHA(I,J)=CACTCO(I,J,2)
10    CONTINUE
      IF (ABS(T-TOLD).LE.1.0E-5) GO TO 52
      N1=NLIGHT+1
      IF(NLIGHT.EQ.0) GO TO 30
C     SET INTERACTION PARAMETER OF NONCONDENSABLE(SOLUTE) WITH THEIR
C     REFERENCE SOLVENT.
      DO 25 K=1,NLIGHT
      NR=NREFER(K)
      GGIJ(K,NR)=DEXP(-ALPHA(K,NR)*GIJ(K,NR))
      GGIJ(NR,K)=0.0
C     SET NONCONDENSABLE-HYPOTHETICAL INTERACTION PARAMETER EQ 1
C     SET NONCONDENSABLE-CONDENSABLE INTERACTION AS SINGLE PARAMETER.
      DO 23 J=N1,NCOMP
      IF((J-NR).EQ.0)GO TO 23
      IF((J-NCRIT).LE.0) THEN
      GGIJ(K,J)=1.0
      GGIJ(J,K)=1.0
      ELSE
      GGIJ(K,J)=DEXP(-ALPHA(K,J)*GIJ(K,J))
      GGIJ(J,K)=0.0
      END IF
23    CONTINUE
C     SET NONCONDENSABLE-NONCONDENSABLE INTERACTION PARAMETER EQUAL 1
```

```
          DO 24 L=1,NLIGHT
            GGIJ(K,L)=1.0
24        CONTINUE
25      CONTINUE

C       CALCULATION OF CONDENSABLE-CONDENSABLE INTERACTION PARAMETER
30      DO 51 I=N1,NCOMP
         DO 51 J=N1,NCOMP
            GGIJ(I,J)=DEXP(-ALPHA(I,J)*GIJ(I,J))
51       CONTINUE

C       CALCULATION OF ACTIVITY COEFF.

52      DO 53 I=1,NCOMP
         R(I)=0.0D0
          DO 53 J=1,NCOMP
           R(I)=R(I)+X(J)*GGIJ(I,J)
53      CONTINUE
        DO 54 I=1,NCOMP
         DO 54 J=1,NCOMP
            XIJ(I,J)=X(J)*GGIJ(I,J)/R(I)
54       CONTINUE
         DO 55 I=1,NCOMP
         SUM1=0.0D0
         SUM2=0.0D0
         SUM3=0.0D0
         DO 56 J=1,NCOMP
            SUM1=SUM1+GIJ(I,J)*GGIJ(I,J)*X(J)
            SUM2=SUM2+GIJ(J,I)*GGIJ(J,I)*XIJ(J,J)
            SUM4=0.0D0
            DO 57 L=1,NCOMP
               SUM4=SUM4+X(L)*GIJ(J,L)*GGIJ(J,L)
57          CONTINUE
            SUM3=SUM3-GGIJ(J,I)*XIJ(J,J)*SUM4/R(J)
56       CONTINUE
        GAMMA(I)=DEXP(SUM1/R(I)+SUM2+SUM3)
55      CONTINUE
        HERT=0.0D0
        DO 100 I=1,NCOMP
          DO 100 J=1,NCOMP
100         HERT=HERT+X(I)*XIJ(I,J)*GIJ(I,J)*(1.0D0+XIJ(I,I)*
     +      ALPHA(I,J)*GIJ(I,J))

        RETURN
        END
```

```
      SUBROUTINE ACTCO9
C     SUBROUTINE TO GET A ACTIVITY COEFFICIENT BY WILSON MODEL
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     + SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     + OMEGA(20),F(20)
      COMMON/PARIP/NCID(20),IP(20,20),NCPM(50,10),NCOMP,NMIX,
     + MIXCOM(50,10),NCM(10)
      COMMON/RST/HENRY(10,30),CHENRY(10,30,2),FREFER(20),
     + ZRA(20),VLIQ(20),VLIQL(20,20),CVLIQL(20,20,2)
      COMMON/ACTCO/CACTCO(20,20,6),GAMMA(20),HERT
      CHARACTER*40 NAME(20)

      DIMENSION XLAMDA(30,30)
      IF (ABS(T-TOLD).LE.1.0E-5) GO TO 55
      RT=1.987*T
      N1=NLIGHT+1
      IF(NLIGHT.EQ.0) GO TO 30
      DO 25 K=1,NLIGHT
        NR=NREFER(K)
        XLAMDA(K,NR)=(VLIQ(NR)/VLIQL(K,NR))*EXP(-CACTCO(K,NR,1)/RT)
        XLAMDA(NR,K)=0.0
        DO 23 J=N1,NCOMP
          IF((J-NR).EQ.0)GO TO 23
          IF((J-NCRIT).LT.0) THEN
            XLAMDA(K,J)=VLIQ(J)/VLIQ(K)
            XLAMDA(J,K)=VLIQ(K)/VLIQ(J)
          ELSE
            IF((CACTCO(K,J,1)).EQ.0) THEN
              XLAMDA(K,J)=XLAMDA(K,NR)*HENRY(K,NR)/HENRY(K,J)
              CACTCO(K,J,1)=-RT*LOG(XLAMDA(K,J)*VLIQL(K,J)/VLIQ(J))
            ELSE
              XLAMDA(K,J)=(VLIQ(J)/VLIQL(K,J))*EXP(-CACTCO(K,J,1)/RT)
              XLAMDA(J,K)=0.0
            END IF
          END IF
23      CONTINUE
        DO 24 L=1,NLIGHT
         XLAMDA(K,L)=VLIQ(L)/VLIQ(K)
24      CONTINUE
25    CONTINUE
30    DO 50 I=N1,NCOMP
      DO 40 J=N1,NCOMP
      XLAMDA(I,J)=(VLIQ(J)/VLIQ(I))*EXP(-CACTCO(I,J,1)/RT)
40    CONTINUE
50    CONTINUE
55    DO 100 I=1,NCOMP
```

```
      XLSUM=0.0
      XSUM1=0.0
      DO 70 J=1,NCOMP
      XLSUM=XLSUM+X(J)*XLAMDA(I,J)
      XSUM2=0.
      DO 60 M=1,NCOMP
      XSUM2=XSUM2+X(M)*XLAMDA(J,M)
60    CONTINUE
      XSUM1=XSUM1+X(J)*XLAMDA(J,I)/XSUM2

70    CONTINUE
      IF((I-NLIGHT).LT.0) THEN
      NR=NREFER(I)
      GAMMA(I)=(XLAMDA(I,NR)/XLSUM)/EXP(XSUM1)
      ELSE
      GAMMA(I)=EXP(1.0-XSUM1)/XLSUM
      END IF
 100  CONTINUE
*
* CALCULATION OF EXCESS ENTHALPHY OF LIQUID MIXTURE
*
      DO 150 I=1,NCOMP
        DO 150 J=1,NCOMP
150         HERT=HERT-X(I)*X(J)*(XLAMDA(I,J)/XLSUM)*
    +       DLOG(VLIQ(I)*XLAMDA(I,J)/VLIQ(J))

      RETURN
      END
      SUBROUTINE ACTCO10

* THIS IS THE SUBROUTINE TO GET ACTIVITY COEFFICIENTS FROM REDLICH-
* KISTER MODEL. THIS SUBROUTINE CAN BE APPLIED ONLY BINARY SYSTEM

      IMPLICIT REAL*8(A-H,O-Z)
      COMMON/COMMON/NAME,P,T,POLD,TOLD,NLIGHT,KV,IMIX,NCRIT,
     + SUMX,SUMY,X(20),Y(20),NREFER(20),VC(20),TC(20,20),PC(20,20),
     + OMEGA(20),F(20)
      COMMON/ACTCO/CACTCO(20,20,6),GAMMA(20),HERT
      CHARACTER*40 NAME(20)
*
* SUBSTITUTE PARAMETER FOR THE GOOD OF PROGRAMMING.

      DO 10 K=1,6
10      CACTCO(2,1,K)=CACTCO(1,2,K)

      B=CACTCO(1,2,1)
      C=CACTCO(1,2,2)
      D=CACTCO(1,2,3)
      E=CACTCO(1,2,4)
      F0=CACTCO(1,2,5)
      G=CACTCO(1,2,6)

      XD=X(1)-X(2)
      XM=X(1)*X(2)


* GET EXCESS GIBBS FREE ENERGY
      GERT=XM*(B+C*XD+D*XD**2+E*XD**3+F0*XD**4+G*XD**5)
      HERT=GERT
* GET ACTIVITY COEFFICIENTS
      P=B+C*XD+D*XD**2+E*XD**3+F0*XD**4+G*XD**5
      Q=C+2*D*XD+3*E*XD**2+4*F0*XD**3+5*G*XD**4

      GAMMA(1)=DEXP(X(2)**2*P+2.0*XM*X(2)*Q)
      GAMMA(2)=DEXP(X(1)**2*P-2.0*XM*X(1)*Q)

      RETURN
      END
```

```
          SUBROUTINE PARA

          CHARACTER*1 A,A1

45        WRITE(*,50)

50        FORMAT(///,' WHICH ONE DO YOU WISH TO CONTROL?'
      +   ///3X,'1) GENERAL PROPERTIES OF EACH CHEMICAL'
      +   //3X,'2) BINARY INTERACTION PARAMETER'
      +   //3X,'3) RETURN TO THE MAIN PROGRAM'
      +   ///3X,' AMONG THREE IF YOU CHOOSE #3,PRESS "RETURN" KEY')

          READ(*,60)A
60        FORMAT(A)
          IF(A.NE.'1'.AND.A.NE.'2') RETURN
65        WRITE(*,70)
70        FORMAT(8X,///,' CHOOSE ONE OF THREE OPTIONS.'
      +   ///3X,'1) FILE CREATION'
      +   //3X,'2) DATA INPUT TO PREVIOUSLY CREATED FILE'
      +   //3X,'3) PARTIAL CORRECTION OF INPUTED DATA'
      +   //3X,'4) RETURN TO THE PREVIOUS STEP')
          READ(*,60)A1
          IF(A1.NE.'1'.AND.A1.NE.'2'.AND.A1.NE.'3') GO TO 45
          IF(A.EQ.'1') THEN
            IF(A1.EQ.'1') CALL GENCRE
            IF(A1.EQ.'2') CALL GENINP
            IF(A1.EQ.'3') CALL GENUP
          ELSE IF(A.EQ.'2') THEN
            IF(A1.EQ.'1') CALL PARACRE
            IF(A1.EQ.'2') CALL PARAINP
            IF(A1.EQ.'3') CALL PARAUP
          END IF
          GO TO 65
          END
```

```
       SUBROUTINE GENCRE
C
C THIS SUBROUTINE IS USED TO CREATE THE NEW FILE WHICH HAS
C GENERAL PROPERTIES OF EACH COMPOUND
C
       IMPLICIT REAL*8 (A-H,O-Z)
      character*40 blank,namei
      CHARACTER*30 FNAME
      CHARACTER*1 A
      DATA BLANK/'                                        '/
* INPUT FILE CONDITION

       WRITE(*,*)' THIS PROGRAM MAY DELETE THE PREVIOUS FILE'
       WRITE(*,*)'PLEASE AVOID USING SAME NAME'

       WRITE(*,*)'FILE NAME?'
       READ(*,10) FNAME
10     FORMAT(A)
       WRITE(*,*)'$ # OF TOTAL MATERIALS?'
       READ(*,*)NTCOMP

30     OPEN(3,FILE=FNAME,STATUS='NEW',
     +ACCESS='DIRECT',FORM='UNFORMATTED',RECL=431)

       DO 100 I=1,NTCOMP
       NAMEI=BLANK
       ANTAI=.0D0
       ANTBI=.0D0
       ANTCI=.0D0
       HARAI=.0D0
       HARBI=.0D0
       HARCI=.0D0
       HARDI=.0D0
       TCI=.0D0
       PCI=.0D0
       VCI=.0D0
       ZCI=.0D0
       OMEGAI=.0D0
       ALQDNI=.0D0
       TDENI=.0D0
       AMWI=.0D0
       TBI=.0D0
       TFI=.0D0
       HVAPI=.0D0
       ZRAI=.0D0
       RI=.0D0
       QI=.0D0
       ZI=.0D0
```

```
        DIPOI=.0D0
        ETAI=0.0D0
        OMGHI=0.0D0
        X1I=0.0D0
        X2I=0.0D0
        X3I=0.0D0
        X4I=0.0D0
        X5I=0.0D0
        X6I=0.0D0
        X7I=0.0D0
        X8I=0.0D0
        X9I=0.0D0
        X10I=0.0D0
        X11I=0.0D0
        X12I=0.0D0
        X13I=0.0D0
        X14I=0.0D0
      WRITE(3,REC=I) NAMEI,ANTAI,ANTBI,ANTCI,HARAI,
     +HARBI,HARCI,HARDI,TCI,PCI,VCI,ZCI,OMEGAI,
     +ALQDNI,TDENI,AMWI,TBI,TFI,HVAPI,ZRAI,RI,QI,ZI,DIPOI,ETAI,OMGHI,
     + X1I,X2I,X3I,X4I,X5I,X6I,X7I,X8I,X9I,X10I,X11I,X12I,X13I,X14I
100   CONTINUE

150   RETURN
      END
```

```
        SUBROUTINE GENINP
C
C      THIS PROGRAM IS FOR DATA INPUT OF FILE NAMED GENCOMP.DAT
C
C
        IMPLICIT REAL*8 (A-H,O-Z)
       character*40 blank,namei
       CHARACTER*30 FNAME
       DATA BLANK/'                                        '/
       WRITE(*,*)'FILE NAME?'
       READ(*,10)FNAME
10     FORMAT(A)

       OPEN(3,FILE=FNAME,STATUS='OLD',
      +ACCESS='DIRECT',FORM='UNFORMATTED',RECL=431)
       K=1
100    WRITE(*,250) K
       READ(*,*) I
       IF(I.EQ.0) GO TO 1
         WRITE(*,110)
110      FORMAT(1X,'READ NAMEI')
       READ(*,300) NAMEI
         WRITE(*,120)
120      FORMAT(1X,'READ ANTAI,ANTBI,ANTCI,HARAI,HARBI,HARCI,HARDI')
       READ(*,*) ANTAI, ANTBI, ANTCI , HARAI,HARBI,HARCI,HARDI
         WRITE(*,130)
130      FORMAT(1X,'READ TCI,PCI,VCI,ZCI,OMEGAI')
       READ(*,*) TCI,PCI,VCI,ZCI,OMEGAI
         WRITE(*,140)
140      FORMAT(1X,'READ ALQDNI,TDENI,AMWI,TBI,TFI')
       READ(*,*) ALQDNI,TDENI,AMWI,TBI,TFI
         WRITE(*,150)
150      FORMAT(1X,'READ HVAPI,ZRAI,RI,QI,ZI,DIPOI,ETAI,OMGHI')
        READ(*,*) HVAPI,ZRAI,RI,QI,ZI,DIPOI,ETAI,OMGHI
         WRITE(*,160)
160      FORMAT(1X,'READ X1I,X2I,X3I,X4I,X5I,X6I,X7I')
         READ(*,*)X1I,X2I,X3I,X4I,X5I,X6I,X7I
         WRITE(*,170)
170      FORMAT(1X,'READ X8I,X9I,X10I,X11I,X12I,X13I,X14I')
         READ(*,*) X8I,X9I,X10I,X11I,X12I,X13I,X14I

       WRITE(3,REC=I) NAMEI,ANTAI,ANTBI,ANTCI,HARAI,
      +HARBI,HARCI,HARDI,TCI,PCI,VCI,ZCI,OMEGAI,
      +ALQDNI,TDENI,AMWI,TBI,TFI,HVAPI,ZRAI,RI,QI,ZI,DIPOI,ETAI,OMGHI
      + X1I,X2I,X3I,X4I,X5I,X6I,X7I,X8I,X9I,X10I,X11I,X12I,X13I,X14I
       K=K+1
       GO TO 100
1      CLOSE(3)
```

```
      RETURN
200   FORMAT(1X,A40,25D10.4)
250   FORMAT(1X,'WRITE THE RECORD # OF COMP',I3,'. IF YOU DO NOT WANT,
     +PRESS 0')
300   FORMAT(A40)
      END
```

```
         SUBROUTINE GENUP

C       THIS IS SUBROUTINE FOR THE PURPOSE OF ACCESSING + UPDATING
C       FILE WHICH CONTAIN GENERAL PROPERTIES OF COMPONENTS

         IMPLICIT REAL*8 (A-H,O-Z)
         CHARACTER*40 NAMEI
         CHARACTER*30 FNAME
         CHARACTER*1 A
         WRITE(*,*)'FILE NAME?'
         READ(*,10)FNAME
10       FORMAT(A)

         OPEN(3,FILE=FNAME,STATUS='OLD',
      +  ACCESS='DIRECT',FORM='UNFORMATTED',RECL=431)
* SELECT THE RECORD # TO BE ACCESSED + UPDATING

30       WRITE(*,*) ' READ THE RECORD # TO BE ACCESSED'
         WRITE(*,*)'   (IF YOU WANT TO FINISH THIS WORK,'
         WRITE(*,*)'    THEN PRESS ANY INTEGER LESS THAN 1)'
         READ(*,*) I

         IF (I.LE.0) THEN
           GO TO 30
         ELSE
* READ PRVIOUSLY STORED DATA

51         READ(3,REC=I,ERR=2) NAMEI,ANTAI,ANTBI,ANTCI,
      +    HARAI,HARBI,HARCI,HARDI,TCI,PCI,VCI,ZCI,
      +    OMEGAI,ALQDNI,TDENI,AMWI,TBI,TFI,HVAPI,ZRAI,
      +    RI,QI,ZI,DIPOI,ETAI,OMGHI,X1I,X2I,X3I,X4I,
      +    X5I,X6I,X7I,X8I,X9I,X10I,X11I,X12I,X13I,X14I

           WRITE(5,200) NAMEI,ANTAI,ANTBI,ANTCI,HARAI,HARBI,
      +    HARCI,HARDI,TCI,PCI,VCI,ZCI,OMEGAI,ALQDNI,
      +    TDENI,AMWI,TBI,TFI,HVAPI,ZRAI,RI,QI,ZI,DIPOI,ETAI,OMGHI,
      +    X1I,X2I,X3I,X4I,X5I,X6I,X7I,X8I,X9I,X10I,X11I,X12I,X13I,
      +    X14I
* DECISION OF UPDATING

           WRITE(*,55)
55         FORMAT(1X,'DO YOU WISH TO UPDATE ?'/1X,'IF YOU WANT CHOOSE Y,
      +    OR NOT CHOOSE N')
           READ(*,60)A
60         FORMAT(A)
           IF (A.EQ.'Y'.OR.A.EQ.'y') THEN
62           WRITE(*,65)
             WRITE(*,70)
```

```
            READ(*,*)N
              IF (N.EQ.0) THEN
                WRITE(3,REC=I)NAMEI,ANTAI,ANTBI,ANTCI,HARAI,HARBI,
     +          HARCI,HARDI,TCI,PCI,VCI,ZCI,OMEGAI,ALQDNI,TDENI,
     +          AMWI,TBI,TFI,HVAPI,ZRAI,RI,QI,ZI,DIPOI,ETAI,OMGHI,
     +          X1I,X2I,X3I,X4I,X5I,X6I,X7I,X8I,X9I,X10I,X11I,X12I,
     +          X13I,X14I
                GO TO 51
              ELSE
* UPDATING THE DATA
                IF (N.EQ.1) THEN
                  WRITE(*,72)
72                FORMAT(1X,'READ NAMEI')
                  READ(*,150) NAMEI
                  GO TO 62
                ELSE
75                WRITE(*,80)
                  READ(*,*)P
                  IF(N.EQ.2)THEN
                    ANTAI=P
                  ELSE IF(N.EQ.3) THEN
                    ANTBI=P
                  ELSE IF(N.EQ.4) THEN
                    ANTCI=P
                  ELSE IF(N.EQ.5) THEN
                    HARAI=P
                  ELSE IF(N.EQ.6) THEN
                    HARBI=P
                  ELSE IF(N.EQ.7) THEN
                    HARCI=P
                  ELSE IF(N.EQ.8) THEN
                    HARDI=P
                  ELSE IF(N.EQ.9) THEN
                    TCI=P
                  ELSE IF(N.EQ.10) THEN
                    PCI=P
                  ELSE IF(N.EQ.11) THEN
                    VCI=P
                  ELSE IF(N.EQ.12) THEN
                    ZCI=P
                  ELSE IF(N.EQ.13) THEN
                    OMEGAI=P
                  ELSE IF(N.EQ.14) THEN
                    ALQDNI=P
                  ELSE IF(N.EQ.15) THEN
                    TDENI=P
                  ELSE IF(N.EQ.16) THEN
                    AMWI=P
                  ELSE IF(N.EQ.17) THEN
                    TBI=P
```

```
    ELSE IF(N.EQ.18) THEN
      TFI=P
    ELSE IF(N.EQ.19) THEN
      HVAPI=P
    ELSE IF(N.EQ.20) THEN
      ZRAI=P
    ELSE IF(N.EQ.21) THEN
      RI=P
    ELSE IF(N.EQ.22) THEN
      QI=P
    ELSE IF(N.EQ.23) THEN
      ZI=P
    ELSE IF(N.EQ.24) THEN
      DIPOI=P
    ELSE IF(N.EQ.25) THEN
      ETAI=P
    ELSE IF(N.EQ.26) THEN
      OMGHI=P
    ELSE IF(N.EQ.27) THEN
      X1I=P
    ELSE IF(N.EQ.28) THEN
      X2I=P
    ELSE IF(N.EQ.29) THEN
      X3I=P
    ELSE IF(N.EQ.30) THEN
      X4I=P
    ELSE IF(N.EQ.31) THEN
      X5I=P
    ELSE IF(N.EQ.32) THEN
      X6I=P
    ELSE IF(N.EQ.33) THEN
      X7I=P
    ELSE IF(N.EQ.34) THEN
      X8I=P
    ELSE IF(N.EQ.35) THEN
      X9I=P
    ELSE IF(N.EQ.36) THEN
      X10I=P
    ELSE IF(N.EQ.37) THEN
      X11I=P
    ELSE IF(N.EQ.38) THEN
      X12I=P
    ELSE IF(N.EQ.39) THEN
      X13I=P
    ELSE IF(N.EQ.40) THEN
      X14I=P
    END IF
  GO TO 62
  END IF
END IF
```

```
              ELSE
                WRITE(*,*)' DO YOU WANT TO UPDATE ANY OTHER MATERIAL?'
                WRITE(*,*)' YES----->PRESS Y OR y, NO---->PRESS ANY KEY'
                READ(*,60)A
                IF(A.EQ.'Y'.OR.A.EQ.'y') GO TO 30
            END IF
          END IF
  2       CLOSE(3)
          RETURN
100       FORMAT(1X,A40,25D10.4)
200       FORMAT(1X,'NAMEI                                        ANTAI
     +  ANTBI        ANTCI'/1X,A40,3(d10.4,1X)/1X,'HARAI        HARBI
     +  HARCI        HARDI       TCI          PCI          VCI'/1X,7(D10.4,1X)/
     +1X,'ZCI          OMEGAI       ALQDNI       TDENI        AMWI         TBI
     +    TFI'/1X,7(D10.4,1X)/1X,'HVAPI        ZRAI         RI           QI
     +      ZI            DIPOI'/1X,6(D10.4,1X)/1X,'ETAI         OMGHI
X1
     +I          X2I          X3I          X4I          '/1X,6(D10.4,1X)/1X,
     + 'X5I         X6I          X7I          X8I          X9I'/1X,5(D10.4,1X)
     +/1X,'X10I        X11I         X12I         X13I         X14I'/1X,
     +5(D10.4,1X))
 65       FORMAT(1X,' NAMEI=1   ANTAI=2  ANTBI=3  ANTCI=4'/
     +1X,'HARAI=5  HARBI=6   HARCI=7  HARDI=8'/1X,'TCI=9   PCI=10
     + VCI=11   ZCI=12  OMEGAI=13   ALQDNI=14'/1X,'TDENI=15  AMWI=16
     + TBI=17   TFI=18,  HVAPI=19  ZRAI=20'/1X,'RI=21 QI=22  ZI=23
     + DIPOI=24 ETAI=25  OMGHI=26'/1X,'X1I=27  X2I=28  X3I=29  X4I=30
     + X5I=31  X6I=32  X7I=33'/1X, 'X8I=34  X9I=35  X10I=36  X11I=37
     + X12I=38  X13I=39  X14I=40')
 70       FORMAT(1X,'CHOOSE THE NUMBER FROM ABOVE,IF NEED NOT CHOOSE 0')
 80       FORMAT(1X,'READ THE VALUE TO BE UPDATED')
150       FORMAT(A40)
          END
```

```
      SUBROUTINE PARACRE

* THIS IS THE SUBROUTINE TO CREATE PARAMETER FILE

      INTEGER RECL,ZERO
      CHARACTER*30 FNAME
      DOUBLE PRECISION VALUE
      DIMENSION X(80)

* PARAMETER CONDITIONS INPUT
      WRITE(*,*)'THIS PROGRAM MAY DELETE THE PREVIOUS FILE'
      WRITE(*,*)'PLEASE AVOID USING THE SAME FILE NAME.'
      WRITE(*,10)
10    FORMAT('$ ENTER DESIRED FILE NAME:')
      READ(*,11)FNAME
11    FORMAT(A)
      WRITE(*,20)
20    FORMAT('$ ENTER DESIRED # OF VARIABLES  PER RECORD:')
      READ(*,*)NVAR
      WRITE(*,30)
30    FORMAT(' $ ENTER THE TOTAL # OF MATERIALS:')
      READ(*,*)MAT
      WRITE(*,40)
40    FORMAT(' $ ENTER DESIRED INITIAL VALUE FOR VARIABLES:')
      READ(*,*)VALUE

* CREATION OF PARAMETER FILE

      OPEN (1,FILE=FNAME,ACCESS='DIRECT',RECL=10*NVAR+1,
     1 FORM='UNFORMATTED',STATUS='NEW')
      DO 100 I=1,MAT
      DO  60 J=1,MAT
      DO  50 K=1,NVAR
      X(K)=VALUE
      IF (J.NE.I) GO TO 50
       X(K)=0.0
50    CONTINUE
      WRITE(1,REC=(I-1)*MAT+J) (X(K),K=1,NVAR)

60    CONTINUE
100   CONTINUE
      CLOSE(1)
      RETURN
      END
      SUBROUTINE PARAINP
C     THIS SUBROUTINE IS MADE FOR THE PURPOSE OF ACCESSING + UPDATING
C     BINARY INTERACTION PARAMETER FILE WHICH CREATED PREVIOUSLY.

      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*30 FNAME
      CHARACTER*1 A
      DIMENSION X(80),NO(80)
* SELECTION OF FILE AND IT'S RECORD # TO BE UPDATED

      WRITE(*,*)' WRITE THE FILE NAME'
      READ(*,10) FNAME
10    FORMAT(A)
      WRITE(*,*)' WRITE THE # OF VARIABLES/RECORD'
      READ(*,*)NVAR
      OPEN(3,FILE=FNAME,STATUS='OLD',ACCESS='DIRECT',FORM=
     + 'UNFORMATTED',RECL=10*NVAR+1)
      WRITE(*,*)' HOW MANY SYSTEMS ARE PRESENT?'
      READ(*,*)NMIX
      DO  200 M=1,NMIX
      WRITE(*,*)' How many components do you select at system #',M
      READ(*,*) NCOMP
      WRITE(*,*)'WRITE THE COMPONENT ID'
      DO 20 I=1,NCOMP
        WRITE(*,*)' COMPONENT #',I,'IS'
        READ(*,*)NO(I)
20    CONTINUE
* DATA INPUT
      DO 40 I=1,NCOMP
        DO 30 J=I,NCOMP
          IF (I.EQ.J) GO TO 30
          WRITE(*,*)'READ',NO(I),'---',NO(J),'INTERACTION VALUE'
          READ(*,*)(X(K),K=1,NVAR)
          WRITE(3,REC=(NO(I)-1)*50+NO(J))(X(K),K=1,NVAR)
          READ(3,REC=(NO(I)-1)*50+NO(J))(X(K),K=1,NVAR)
          WRITE(*,25)(X(K),K=1,NVAR)
          WRITE(*,*)'READ',NO(J),'---',NO(I),' INTERACTION VALUE'
          READ(*,*)(X(K),K=1,NVAR)
          WRITE(3,REC=(NO(J)-1)*50+NO(I))(X(K),K=1,NVAR)
          READ(3,REC=(NO(J)-1)*50+NO(I))(X(K),K=1,NVAR)
          WRITE(*,25)(X(K),K=1,NVAR)
25        FORMAT(1X,6(1X,D10.4)/1X,6(1X,D10.4)/1X,6(1X,D10.4))
30      CONTINUE
40    CONTINUE
200   CONTINUE
      RETURN
      END
      SUBROUTINE PARAUP

C     THIS PROGRAM IS MADE FOR THE PURPOSE OF ACCESSING + UPDATING
C     BINARY INTERACTION PARAMETER FILE WHICH CREATED PREVIOUSLY.

      IMPLICIT REAL*8 (A-H,O-Z)
      CHARACTER*30 FNAME
      CHARACTER*1 A
      DIMENSION X(80)

* PARAMETER FILE CONDITIONS INPUT

      WRITE(*,*)' WRITE THE FILE NAME'
      READ(*,10) FNAME
10    FORMAT(A)
      WRITE(*,*)' WRITE THE # OF VARIABLES/RECORD'
      READ(*,*) NVAR
      OPEN(3,FILE=FNAME,STATUS='OLD',ACCESS='DIRECT',FORM=
     +'UNFORMATTED',RECL=10*NVAR+1)
5     WRITE(*,*)' WRITE THE # OF 1ST MATERIAL'
      WRITE(*,*)' IF YOU DO NOT WISH TO CHOOSE PRESS 0'
      READ(*,*)J
      WRITE(*,*)' WRITE THE # OF 2ND MATERIAL'
      WRITE(*,*)' IF YOU DO NOT WISH TO CHOOSE PRESS 0'
      READ(*,*)K
      IF ((J.EQ.0).OR.(K.EQ.0)) RETURN
30    READ(3,REC=(J-1)*50+K)(X(M),M=1,NVAR)
      WRITE(*,45)(X(M),M=1,NVAR)
45    FORMAT(1X,10(1X,D10.4))
46    WRITE(*,*)' DO YOU WISH TO UPDATE?'
      WRITE(*,*)' YES---> Y,NO---> N'
      READ(*,10)A
      IF (A.EQ.'N'.OR.A.EQ.'n') GO TO 68
      IF(A.EQ.'Y'.OR.A.EQ.'y') GO TO 49
      WRITE(*,*)' YOU ENTER THE WRONG VALUE'
      GO TO 46
49    IF (NVAR.EQ.1) THEN
        WRITE(*,*)' WRITE THE NEW VALUE TO BE UPDATED'
        READ(*,*)P
        X(NVAR)=P
      ELSE
        WRITE(*,*)' CHOOSE THE # OF VARIABLE,IF NEED NOT PUSH 0'
        READ(*,*)N
        IF(N.EQ.0) RETURN
        WRITE(*,*)' WRITE THE NEW VALUE TO BE UPDATED'
        READ(*,*)P
        X(N)=P
      END IF
      WRITE(3,REC=(J-1)*50+K)(X(M),M=1,NVAR)
      READ(3,REC=(J-1)*50+K)(X(M),M=1,NVAR)
      WRITE(*,65)(X(M),M=1,NVAR)
65    FORMAT(1X,10(1X,D10.4))

68    WRITE(*,*)'DO YOU WISH TO UPDATE OTHERS?'
      WRITE(*,*)'YES--->Y,NO---> PRESS ANY KEY'
      READ(*,10)A
      IF (A.EQ.'Y'.OR.A.EQ.'y') GO TO 5

      RETURN
      END
```