A MATHEMATICAL MODEL OF REACTIVE DISTILLATION FOR ETHYL ACETATE SYNTHESIS

WALAIRAT SUKSAMAI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN PETROCHEMICAL ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2002

ISBN 974-648-861-9

หัวข้อวิทยานิพนธ์	แบบจำลองทางคณิตศาสตร์ของการกลั่นรีแอกทีฟสำหรับการสังเคราะห์เอธิลอะซิเตต
นักศึกษา	นางสาววลัยรัตน์ สุขสมัย
รหัสประจำตัว	42061204
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมปิโตรเคมี
พ.ศ.	2545
อาจารย์ผู้ควบคุมวิทยานิพนธ์	ผศ.ดร. ประกอบ กิจไชยา

บทคัดย่อ

งานวิจัยนี้ นำเสนอแบบจำลองทางคณิตศาสตร์ซึ่งใช้ทำนายพฤติกรรมที่คาดว่าจะเกิดขึ้นในหอกลั่นรีแอกทีฟ โดยทำการศึกษากับปฏิกิริยาเอกพันธ์ในวัฏภาคของเหลว ในที่นี้ศึกษาปฏิกิริยาเอสเทอริฟิเคชันระหว่างกรดอะซิติกกับเอธานอล ภายใต้สมมุติฐานของแบบจำลองแบบไม่สมดุล โดยใช้วิธีการยูนิแฟกในการทำนายค่าคงที่สมดุลของไอ-ของเหลว และใช้วิธีการหาค่าต่ำสุดของผลรวมยกกำลังสองของฟังก์ชันความผิดพลาดในการแก้ระบบสมการของแบบจำลอง นำผลของแบบจำลองที่ได้ไปเปรียบเทียบกับผลการทดลองและผลที่ได้จากแบบจำลองแบบสมดุล พบว่าผลของแบบจำลองแบบไม่สมดุลใกล้เคียงกับผลการทดลองมากกว่าผลของแบบจำลองแบบสมดุล

Thesis Title A Mathematical Model of Reactive Distillation for Ethyl Acetate Synthesis
Student Miss Walairat Suksamai
Student ID. 42061204
Degree Master of Engineering
Programme Petrochemical Engineering
Year 2002
Thesis Advisor Asst. Prof. Dr. Prakob Kitchaiya

ABSTRACT

This thesis proposes a mathematical model for predicting behaviors inside a reactive distillation column where a homogeneous liquid phase reaction taking place, e.g., an esterification of acetic acid with ethanol. This model would be based on a nonequilibrium. The UNIFAC method is applied to predict vapor-liquid equilibrium constants. A method of minimization of sum of square error is used to solve the set of model equations. The obtained results are compared with the experimental results and the simulation results which based on an equilibrium model. It is found that the simulation results of the nonequilibrium model agree satisfactorily with the experimental results and it provides the better results comparing with those of the equilibrium model.

ACKNOWLEDGEMENTS

This thesis would have never been accomplished without the generous contribution from many people. Firstly, I would like to thank my advisor, Asst. Prof. Dr. Prakob Kitchaiya, for his valuable advice, kindness and great patience.

Special thanks to Mr. Surat Areerat for finding the Komatsu papers. I am thankful to everyone in the chemical engineering department for their help.

Finally, I am very grateful for my family who supports and encourages me every time, especially during this study.

Walairat Suksamai

TABLE OF CONTENTS

TABLE OF CONTENTS (CONTINUED)

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

a_i	constant in ideal gas heat capacity equation
a_j	effective interfacial area of the jth tray (m^2/tray)
a_{mk}	group interaction parameter in UNIFAC method
b_i	constant in ideal gas heat capacity equation
c_i	constant in ideal gas heat capacity equation
C	total number of components
C_i	concentration of component i in the liquid phase (kmol/m^3)
C_p^L	heat capacity of liquid (J/mol K)
C_p^V	heat capacity of vapor (J/mol K)
C_{pM}^L	average heat capacity of liquid (J/mol K)
C_{pM}^V	average heat capacity of vapor (J/mol K)
d_i	constant in ideal gas heat capacity equation
D	liquid flow rate of the distillate (mol/s)
D_{ik}^L	Fick's binary diffusion coefficient in the liquid phase (cm^2/s)
D_{ik}^V	Fick's binary diffusion coefficient in the vapor phase (cm^2/s)
E_1	residual function for heat balance in the condenser
E_N	residual function for heat balance in the reboiler
E_j^I	residual function for heat transfer rate from the vapor phase to the liquid phase on the jth tray
E_j^L	residual function for heat balance in the liquid phase on the jth tray
E_j^V	residual function for heat balance in the vapor phase on the jth tray
f_{ij}^L	liquid feed rate of component i on the jth tray (mol/s)
f_{ij}^V	vapor feed rate of component i on the jth tray (mol/s)
F_S	superfacial F-factor ($kg^{0.5}/m^{0.5}$ s)
h_j^L	heat transfer coefficient in the liquid phase (MJ/m^2 s K)
h_j^V	heat transfer coefficient in the vapor phase (MJ/m^2 s K)
H_j^L	molar enthalpy of liquid (MJ/mol)
H_j^V	molar enthalpy of vapor (MJ/mol)
$\tilde{H}_{ij}^L$	partial molar enthalpy of component i in the liquid on the jth tray (MJ/mol)

H_j^{LF}	molar enthalpy of liquid feed (MJ/mol)
$\tilde{H}_{ij}^{V}$	partial molar enthalpy of component *i* in the vapor on the jth tray (MJ/mol)
H_j^{VF}	molar enthalpy of vapor feed (MJ/mol)
k	thermal conductivity (kW/m K)
k_{av}^{L}	average multicomponent mass transfer coefficient in the liquid phase (kmol/m^2 s)
k_{av}^{V}	average multicomponent mass transfer coefficient in the vapor phase (kmol/m^2 s)
k_{ik}^{L}	mass transfer coefficient of binary i-k pare in the liquid phase (mol/m^2 s)
k_{ik}^{V}	mass transfer coefficient of binary i-k pare in the vapor phase (mol/m^2 s)
κ_{ik}^{L}	matrix element of multicomponent mass transfer coefficients at the liquid side of the interface (mol/m^2 s)
κ_{ik}^{V}	matrix element of multicomponent mass transfer coefficients at the vapor side of the interface (mol/m^2 s)
K_{iN}	vapor-liquid equilibrium constant of component i in the reboiler
K_{ij}^{I}	vapor-liquid equilibrium constant at the interface on the jth tray
l_i	constant in UNIFAC method
Le	Lewis number
L_F	volumetric liquid flow rate per unit weir length (m^2/s)
L_j	total liquid flow rate from the jth tray (mol/s)
L_1	liquid flow rate from the condenser to the first top tray in a column (mol/s)
L_N	bottom liquid flow rate (mol/s)
M	number of independent variables
M_{i1}	residual function for component material balance in the condenser
M_{iN}	residual function for component material balance in the reboiler
N	total number of trays including the condenser and the reboiler
N_{ij}	mass transfer rate for component *i* from the vapor to the liquid phase (mol/s)
P	column pressure (atm)
Pr	Prandtl number
P^{sat}	vapor pressure (atm)
q_i	relative surface area of a molecule in UNIFAC method
Q	Number of random points

Q_C	heat duty of the condenser (W)
Q_k	area parameter of functional group k in UNIFAC method
Q_R	heat duty of the reboiler (W)
Q_j^L	heat duty in the liquid phase on the jth tray (W)
Q_j^V	heat duty in the vapor phase on the jth tray (W)
Q_{iN}	residual function for phase equilibrium relation for the reboiler
Q_{ij}^I	residual function for phase equilibrium relation at the interface on the jth tray
r_i	relative number of segments per molecule in UNIFAC method
r_{ij}^L	reaction rate of component i on the jth tray (mol/m^3 s)
R_i	a random number distributed uniformly on the interval (0,1)
R_k	volume parameter for functional group k in UNIFAC method
R_{ij}^L	residual function for mass transfer rate from the interface to the liquid phase
R_{ij}^V	residual function for mass transfer rate from the vapor phase to the interface
Sc	Schmidt number
S_j^L	ratio of withdrawal to liquid flow rate of inter-tray
S_j^V	ratio of withdrawal to vapor flow rate of inter-tray
t_L	average liquid contact time (s)
T_C	critical temperature (K)
T_{mk}	binary interaction parameter in UNIFAC method
T_r	reduced temperature
T_j^I	temperature at the interface on the jth tray (K)
T_j^L	temperature of bulk liquid phase on the jth tray (K)
T_j^V	temperature of bulk vapor phase on the jth tray (K)
U_j	volumetric liquid holdup on the jth tray (l)
V_j	total vapor flow rate from the jth tray (mol/s)
W_x	exit weir height (m)
x_{ij}	mole fraction of component i in the bulk liquid on the jth tray
x_{ij}^I	mole fraction of component i at the liquid side of the interface on the jth tray
x^R	the rejected point
$x^{(L)}$	variable lower bounds
$x^{(U)}$	variable upper bounds

$\bar{x}$	centroid of random points
X_m	mole fraction of functional group m in UNIFAC method
y_{ij}	mole fraction of component i in the bulk vapor on the jth tray
y_{ij}^{I}	mole fraction of component i at the voper side of the interface on the jth tray
Z_{C}	clear liquid height (m)
Z_L	average liquid flow path length (m)
$\bar{Z}$	lattice coordination number in UNIFAC method
α	parameter whish determines the distance of the reflection
Γ_k	residual activity coefficient of the functional group k in the actual mixture
$\Gamma_k^{(i)}$	residual activity coefficient of the functional group k in the reference mixture
Γ_{ik}^{L}	thermodynamic function defined by (3.52)
γ_i	activity coefficient of component i in the mixture
γ_i^{C}	activity coefficient in the combinatorial part
γ_i^{R}	activity coefficient in the residual part
θ_i	area fraction in UNIFAC method
λ_i	latent heat of vaporization of component i
μ	viscosity (cP)
$\nu_k^{(i)}$	number of functional groups of type k in molecule i
ρ^{V}	molar density of vapor (kmol/m^3)
ψ_i	segment fraction in UNIFAC method

CHAPTER 1

INTRODUCTION

1.1 Statement and significant of the problems

In the chemical processing industries, there are several important processes combined in sequences. Two of them are a reaction process where chemical reactions take place and a separation process where the products are purified. The reactive distillation is an efficient process unit that combines these two important processes in a single unit operation. Due to one process unit is eliminated; capital and operating costs are reduced. Additionally, there are any other advantages associated with this combination unit. The advantages include improving product yield and selectivity, overcoming equilibrium limitations on conversion and limitations of azeotropic mixtures, suppressing side reactions to reduce undesired products, saving energy for exothermic reaction because the reaction heat is used to vaporize the liquid for distillation, and simplifying temperature control of the reaction system.

For the above reasons, the reactive distillation has received increasing interest. This research focuses on setting a mathematical model to simulate the reactive distillation column where the esterification of acetic acid with ethanol occurs. This model bases on the assumption of steady-state nonequilibrium stage or rate-based model. UNIFAC method is applied to predict vapor-liquid equilibrium relations.

The simulations can help studying and developing reactive distillation unit both in design and control basis.

1.2 Goal and objective

The goal and objective of the research was to study the behaviors inside a multicomponent reactive distillation by simulating and compare the simulated results with experimental data.

1.3 Hypothesis

1. When comparing with the experimental data of a reactive distillation system, a nonequilibrium model may provide better results than an equilibrium model.
2. In reactive distillation problems, especially in a nonideal system, the model using the UNIFAC method for predicting vapor-liquid equilibrium constants may give better results comparing with those of the empirical formula.

1.4 Scope of the study

The research goals and objectives were as follow:

1. To set a mathematical model of a reactive distillation column where the esterification of acetic acid with ethanol, a homogeneous liquid phase reaction, takes place.
2. To apply the UNIFAC method for predicting vapor-liquid equilibrium data in a reactive distillation system.
3. To compare the calculated results of the model with the experimental results.

1.5 Process of the study

The processes of the study were:

1. reviewing the related literatures
2. assigning the assumption
3. setting the mathematical model
4. solving the model equations
5. comparing results with the experimental data and discussion

CHAPTER 2

LITERATURE REVIEW

The concept of using reactive distillation technique was first applied in the early 1920's when Backhause granted a series of U.S. patents concerning esterification process (DeGarmo et al., 1992) and was reviewed by Keyes(1932) in 1932. Although the theoretical basis of reactive distillation was carried out by Leyes and Othmer (1945), because of the substantial difficulty of solving an exact design problem, the manual plate-to-plate calculations with very rough approximations in vapor-liquid equilibrium and reaction rate relations were used to solve it.

Until the 1970's, the rigorous mathematical models for computer simulation were turn out to be developed. Previously, the models always based on the assumption of the steady-state with the equilibrium model by using simple thermodynamic and reaction rate relations for solving the problems.

Suzuki et al.(1971) applied modified Muller's method, an iterative method, for determining stage temperature, stage reaction rates and interstage flow rates of multicomponent reactive distillation system.

Komatsu (1977) developed the relaxation method, the new method for correcting liquid compositions. His calculation results were compared with the actual values given by the experimental data.

Sawistowski and Pilavkis (1979) modeled a packed reactive distillation column for the esterification of acetic acid with methanol. They used an effective diffusivity method for their mass transfer model. It was the starting point of using the modification of the nonequilibrium model of Krishnamurthy and Taylor (1985a, b, c) to handle reactive distillation.

The Naphtali-Sandholm method, the method for cracking a set of nonlinear equation which was lack of convergence and needed a large computer memory, was used by Murthy (1984) for solving the model of distillation columns in which chemical reactions occur.

Grosser et al. (1987) studied and proposed the model of reactive distillation in Nylon 6,6 production under the assumption of unsteady or dynamic model and developed guidelines for the use of reactive entrainers to ease the separation of closely boiling mixtures.

Alejski et al. (1988) applied Powell's method, a minimization method, to solve reactive distillation problems. This approach was, however, quite slow to converge. Their numerical example was the esterification of acetic acid with ethanol. They compared the computed result with the experimental data from Komatsu (1977). Chang and Seader (1988) developed and demonstrated a robust numerical procedure for reactive distillation, involving the use of homotopy-continuation in the manner of Wayburn and Seader (1988). This procedure was designed to permit great flexibility with respect to column configuration and choice of thermodynamic properties. Barbosa and Doherty (1988) studied the influence of equilibrium chemical reactions on vapor-liquid phase diagrams, derived a set of ordinary differential equations describing the dynamics of homogeneous reactive distillation and presented a general method of calculating minimum reflux ratios for single-feed multicomponent reactive distillation columns.

Bogacki et al. (1989) proposed the Adams-Moulton method for solving a reactive distillation problem both for steady-state and unsteady state. This proposed method was very fast and numerically stable therefore no corrected techniques were needed. The obtained results were consistent with the experimental data of Komatsu (1977) and the computed results of Alejski et al. (1989).

Agreda et al. (1990) presented their article described the development of Eastman Kodak's commercial process for the production of high-purity methyl acetate via countercurrent reactive distillation. This development and scale-up steps included concept inception, laboratory testing, bench-scale testing, pilot plant development, modeling and design, full-scale construction and operation, and final troubleshooting and optimization. Fuchigimi (1990) reported top to bottom column temperature and composition data of 13 experiments for the methyl acetate process carried out in a laboratory-scale column.

Duprat and Gau (1991) studied the separation of close-boiling point mixtures using reactive distillation. Their work was focused on the separation of 3-picoline and 4-picoline through complexation by trifluroacetic acid. Newton's method was used to solve the equilibrium stage model equations. Simandl and Svrchek (1991) applied two algorithm, simultaneous solution after linearization and the inside-outside method for solving the equations describing the steady state operation of reactive distillation column. They showed that the shape of the profiles obtained by the equilibrium model is largely in agreement with shape of the profiles measured by Komatsu et al. (1970), even though there were some significant quantitative differences.

Zheng and Xu (1992) studied the mass transfer of gas-liquid and liquid-catalyst granules and derived correlation for computing liquid-solid mass transfer coefficients in the catalyst bed. They used this correlation with a nonequilibrium model to simulate catalytic distillation operations in a packed column with a bag-type porous catalyst.

Jacobs and Krishna (1993) applied a steady-state equilibrium stage model to simulate a reactive distillation column for the synthesis of methyl tert-butyl ether by using the rigorous distillation model, REDFRAC, from a flowsheet simulator namely Aspen Plus. Bravo and Pyhalahti (1993) provided some preliminary results obtained from a catalytic distillation pilot plant used to produce tert-amyl methyl ether. Kerkhof and Mak (1993) presented results of computer simulations of methyl tert-butyl ether in a fixed-bed reactor and in a reactive distillation column. From the simulations, they concluded that it might be feasible to install more than one methanol feed location on a column in order to avoid an unwanted steady-state caused by column misoperation.

Sundmacher et al. (1994) derived nine dimensionless parameter groups which characterized the mass and heat transfer phenomena between the liquid, the vapor and the solid phase in the reactive distillation process. These groups were applied to analyze industrial processes and experimental data for esterification, etherification and alkylation regarding different catalysts, operation conditions and configurations of plants. The number of each group gave information about how to select adequate models, how to set up better models and how to increase performance of reactive distillation units by mean of developing new catalysts.

Abufares and Douglas (1995) applied an equilibrium state model for both steady-state and dynamic modeling of a methyl tert-butyl ether reactive distillation column. In their work, SpeedUp, an equation-oriented simulation package, was used to solve the steady-state and dynamic equations describing the system behavior and compared the steady-state results with the results predicted by using Aspen Plus.

Alejski and Duprat (1996) studied the problems of reactive distillation simulation with regard to a mathematical model formulation in case of kinetically controlled reactions and showed the main factors which determined definite dynamic behavior of the column.

Lee and Dudukovic (1998) described a nonequilibrium stage model for reactive distillation column which the homogeneous liquid phase reversible reaction between acetic acid and ethanol in production of ethyl acetate took place The results were compared with those of equilibrium stage model. The binary mass transfer coefficients were predicted from the empirical correlation of the AIChE method and The Maxwell-Stefan equations were used to describe the interface transport. Newton's method and homotopy continuation were applied to solve the model equations. A close agreement between predictions for equilibrium and nonequilibrium models was found only when the tray efficiency could be correctly predicted for the equilibrium model.

Taylor and Krishna (2000) traced the development of models that have been used for design of reactive distillation columns and suggested future research direction.

CHAPTER 3

THEORETICAL CONSIDERATION

This chapter would explain the mathematical models which set based on both of an equilibrium and nonequilibrium model including relating assumptions and those relations that being used for modeling parameter calculations.

3.1 Mathematical model

The reactive distillation column is similar to that of a typical distillation column as shown in figure 3.1.

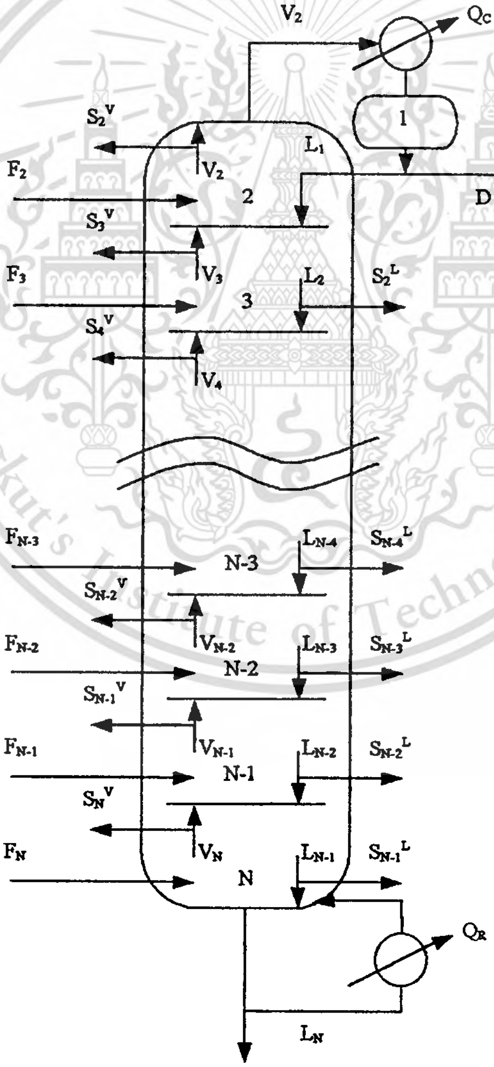

Figure 3.1 Scheme of the reactive distillation column

There are three major parts in this system namely total condenser, tower and partial reboiler. The model equations of both equilibrium-based and nonequilibrium-based that would represent physicochemical phenomena in the above three parts will be shown shortly.

3.1.1 **Equilibrium model** (Lee & Dudukovic, 1998; Seader & Henley, 1998)

Assumptions

1 The process has reached a continuous steady-state.

2 The vapor and liquid bulks are perfectly mixed and are at thermodynamic equilibrium.

3 Heat of mixing is neglected.

4 The reaction occurs only in the liquid bulk.

5 The heat losses from the column walls are ignored.

6 The column hydraulics is neglected.

The subscript representing tray number of any terms in this part and the next part would be assigned as j = 1, j = 2,3,..., N -1 and j = N, for the total condenser, the column, and the partial reboiler, respectively. The mathematical model is as follows:

3.1.1.1 Total condenser (j = 1)

Material balance for component i

$$M_{i1} = (1+(1/R))x_{i1}L_1 - y_{i2}V_2 - r_{i1}^L U_1 = 0, \quad i = 1,2,\ldots,C \quad (3.1)$$

Energy balance

$$E_1 = (1+(1/R))H_1^L L_1 + Q_C - H_2^V V_2 = 0 \quad (3.2)$$

Mole fraction constraint equation of liquid

$$\sum_{i=1}^{C} x_{i1} - 1 = 0 \quad (3.3)$$

3.1.1.2 Equilibrium stages (j = 2, 3,..., N -1)

Material balance for component i

$$M_{ij} = x_{ij}L_j + y_{ij}V_j - x_{ij-1}L_{j-1} - y_{ij+1}V_{j+1} - f_{ij}^L - r_{ij}^L U_j = 0, \quad i = 1,2,\ldots,C \tag{3.4}$$

Energy balance

$$E_j = H_j^V V_j + H_j^L L_j - H_{j+1}^V V_{j+1} - H_{j-1}^L L_{j-1} - H_j^{LF}\sum_{i=1}^{C} f_{ij}^L = 0 \tag{3.5}$$

Phase equilibrium relation

$$Q_{ij} = K_{ij}x_{ix} - y_{ij} = 0, \quad i = 1,2,\ldots,C \tag{3.6}$$

Mole fraction constraint equation in vapor phase

$$\sum_{i=1}^{C} y_{ij} - 1 = 0 \tag{3.7}$$

Mole fraction constraint equation in liquid phase

$$\sum_{i=1}^{C} x_{ij} - 1 = 0 \tag{3.8}$$

3.1.1.3 Partial reboiler (j = N)

Material balance for component i

$$M_{iN} = y_{iN}V_N + x_{iN}L_N - x_{iN-1}L_{iN-1} - r_{iN}^L U_N = 0, \quad i = 1,2,\ldots,C \tag{3.9}$$

Energy balance

$$E_N = H_N^V V_N + H_N^L L_N - H_{N-1}^L L_{N-1} - Q_R = 0 \tag{3.10}$$

Phase equilibrium relation

$$Q_{iN} = K_{iN} x_{iN} - y_{iN} = 0, \; i = 1,2,\ldots C \tag{3.11}$$

Mole fraction constraint equation in vapor phase

$$\sum_{i=1}^{C} y_{iN} - 1 = 0 \tag{3.12}$$

Mole fraction constraint equation in liquid phase

$$\sum_{i=1}^{C} x_{iN} - 1 = 0 \tag{3.13}$$

3.1.2 Nonequilibrium model (Lee & Dudukovic, 1998; Seader & Henley, 1998)

Assumptions

1. The process has reached a continuous steady-state.
2. The vapor and liquid bulks on each side of the interface are perfectly mixed.
3. The vapor-liquid interface is uniform in each tray and is at thermodynamic equilibrium.
4. Heat of mixing is neglected.
5. There is no accumulation of mass and heat at the interface.
6. The reaction occurs only in the liquid bulk.
7. The heat losses from the column walls are ignored.
8. The column hydraulics is neglected.

The mathematical model is as follow:

3.1.2.1 Total Condenser (j = 1)

Material balance for component i

$$M_{i1} = (1+(1/R))x_{i1}L_1 - y_{i2}V_2 - r_{i1}^{L}U_1 = 0, \quad i=1,2,..,C \qquad (3.14)$$

Energy balance

$$E_1 = (1+(1/R))H_1^{L}L_1 + Q_C - H_2^{V}V_2 = 0 \qquad (3.15)$$

Mole fraction constraint equation of liquid

$$\sum_{i=1}^{C} x_{i1} - 1 = 0 \qquad (3.16)$$

3.1.2.2 Nonequilibrium stages (j = 2, 3,..., N -1)

A schematic diagram of nonequilibrium stage is illustrated in figure 3.2.

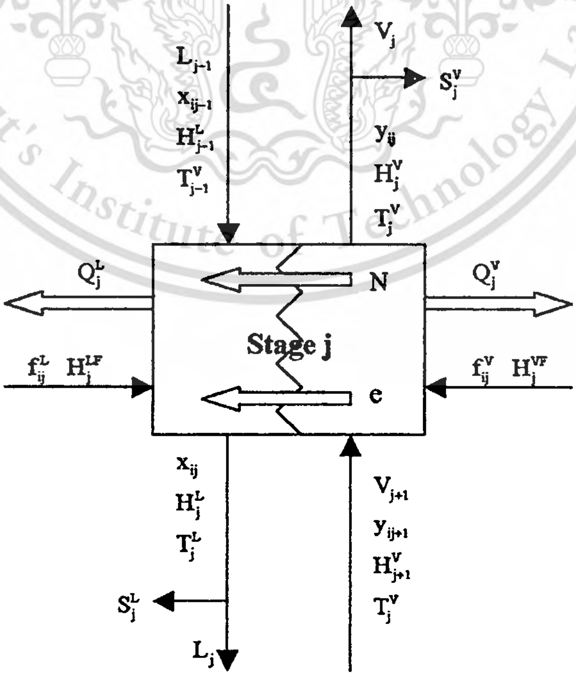

Figure 3.2 Schematic diagram of nonequilibrium stage j

Total vapor-phase material balance

$$M^V_{Tj} = \left(1+S^V_j\right)V_j - V_{j+1} - \sum_{i=1}^{C} f^V_{ij} - N_{T,j} = 0 \tag{3.17}$$

Total liquid-phase material balance

$$M^L_{Tj} = \left(1+S^L_j\right)L_j - L_{j-1} - \sum_{i=1}^{C} f^L_{ij} - N_{T,j} = 0 \tag{3.18}$$

Vapor-phase material balance for component i on the jth tray

$$M^V_{ij} = \left(1+S^V_j\right)y_{ij}V_j - y_{ij+1}V_{j+1} - f^V_{ij} + N_{ij} = 0, \quad i = 1.2,\ldots,C \tag{3.19}$$

Liquid-phase material balance for component i on the jth tray

$$M^L_{ij} = \left(1+S^L_j\right)x_{ij}L_j - x_{ij-1}L_{j-1} - f^L_{ij} - N_{ij} - r^L_{ij}U_j = 0, \quad i = 1,2,\ldots,C \tag{3.20}$$

To explain the behavior at the interface, it could be postulated, as shown in figure 3.3, that a thin vapor film exists on one side of the interface and a thin liquid film exists on the other side.

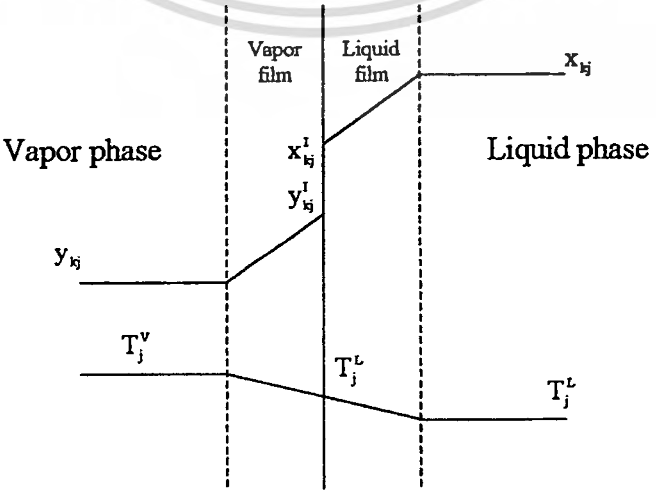

Figure 3.3 Schematic representation of a nonequilibrium tray

Vapor-phase energy balance on the jth tray

$$E_j^V = \left(1+S_j^V\right)H_j^V V_j + Q_j^V + h_j^V a_j\left(T_j^V - T_j^I\right) - H_{j+1}^V V_{j+1} - H_j^{VF}\sum_{i=1}^{C} f_{ij}^V + \sum_{i=1}^{C} N_{ij}\widetilde{H}_{ij}^V = 0 \qquad (3.21)$$

Liquid-phase energy balance on the jth tray

$$E_j^L = \left(1+S_j^L\right)H_j^L L_j + Q_j^L - h_j^L a_j\left(T_j^I - T_j^L\right) - H_{j-1}^L L_{j-1} - H_j^{LF}\sum_{i=1}^{C} f_{ij}^L + \sum_{i=1}^{C} N_{ij}\widetilde{H}_{ij}^L = 0 \qquad (3.22)$$

Vapor-phase mass transfer for component i within the interface on the jth tray

$$R_{ij}^V = N_{ij} - \sum_{k=1}^{C-1} \kappa_{ik}^V a_j\left(y_{kj} - y_{kj}^I\right) - y_{ij}\sum_{k=1}^{C} N_{kj} = 0 \qquad (3.23)$$

where κ_{ik}^V is binary-pair coefficients in the vapor bulk defined in the section below.

Liquid-phase mass transfer for component i within the interface on the jth tray

$$R_{ij}^L = N_{ij} - \sum_{k=1}^{C-1} \kappa_{ik}^L a_j\left(x_{kj}^I - x_{kj}\right) - x_{ij}\sum_{k=1}^{C} N_{kj} = 0 \qquad (3.24)$$

where κ_{ik}^L is binary-pair coefficients in the liquid bulk defined in the section below.

Energy transfer within the interface on the jth tray

$$E_j^I = h_j^V a_j\left(T_j^V - T_j^I\right) + \sum_{i=1}^{C} N_{ij}\widetilde{H}_{ij}^V - h_j^L a_j\left(T_j^I - T_j^L\right) - \sum_{i=1}^{C} N_{ij}\widetilde{H}_{ij}^L = 0 \qquad (3.25)$$

Phase equilibrium relation at the interface on the jth tray

$$Q^{I}_{ij} = K^{I}_{ij}x^{I}_{ij} - y^{I}_{ij} = 0 \qquad (3.26)$$

Mole fraction constraint equation of liquid at the interface

$$\sum_{i=1}^{C} x^{I}_{ij} = 0 \qquad (3.27)$$

Mole fraction constraint equation of vapor at the interface

$$\sum_{i=1}^{C} y^{I}_{ij} = 0 \qquad (3.28)$$

Mole fraction constraint equation of the bulk liquid

$$\sum_{i=1}^{C} x_{ij} = 0 \qquad (3.29)$$

Mole fraction constraint equation of the bulk vapor

$$\sum_{i=1}^{C} y_{ij} = 0 \qquad (3.30)$$

3.1.2.3 Partial reboiler ($j = N$)

Material balance for component i

$$M_{iN} = y_{iN}V_N + x_{iN}L_N - x_{iN-1}L_{N-1} - r^{L}_{iN}U_N = 0, \quad i = 1,2,\ldots,C \qquad (3.31)$$

Energy balance

$$E_N = H^{V}_{N}V_N + H^{L}_{N}L_N - H^{L}_{N-1}L_{N-1} - Q_R = 0 \qquad (3.32)$$

Phase equilibrium relation

$$Q_{iN} = K_{iN}x_{iN} - y_{iN} = 0, \qquad i = 1,2,\dots,C \tag{3.33}$$

Mole fraction constraint equation in vapor phase

$$\sum_{i=1}^{C} y_{iN} - 1 = 0 \tag{3.34}$$

Mole fraction constraint equation in liquid phase

$$\sum_{i=1}^{C} x_{iN} - 1 = 0 \tag{3.35}$$

3.2 Transport-rate expressions and transport properties

Nonequilibrium-based or rate-based models use the same vapor-liquid equilibrium relation and enthalpy correlations as equilibrium-based models. However, the K-values apply only at the equilibrium interface between the vapor and liquid phases on trays or in packing. In general, the vapor-liquid equilibrium relation, whether based on an equation-of-state or an activity-coefficient model, is a function of phase-interface temperature and compositions, and tray pressure. Enthalpies are evaluated only at the conditions of phases as they exit a tray. For the equilibrium-based model, the vapor is at the dew-point temperature and the liquid is at the bubble-point temperature, where both temperatures are equal and at the stage temperature. For the rate-based model, the liquid is subcooled and the vapor is superheated.

The accuracy of enthalpies and, particularly, the vapor-liquid equilibrium relation is crucial to equilibrium-based models, In addition, accurate predictions of heat-transfer rates and particularly mass-transfer rates are required for rate-based models. These rates depend on transport coefficients, interfacial area, and driving forces. It is important that mass-transfer rates account for component-coupling effects through binary-pair coefficients.

3.2.1 Mass transfer rate (Seader & Henley, 1998)

The general forms for component mass transfer rate across the vapor and liquid films, respectively, on a tray or in a packed segment, are as follows, where both diffusive and convective (bulk-flow) contributions are included:

$$N_{i,j}^{V} = a_j^I J_{i,j}^{V} + y_{i,j} N_{T,j} \tag{3.36}$$

and

$$N_{i,j}^{L} = a_j^I J_{i,j}^{L} + x_{i,j} N_{T,j} \tag{3.37}$$

where a_j^I is the total interfacial area for the stage and $J_{i,j}^P$ is the molar diffusion flux relative to molar average velocity, where P stands for the phase (V or L). For a binary mixture, these fluxes, in terms of mass-transfer coefficients, are given by

$$J_i^V = c_t^V k_i^V \left(y_i^V - y_i^I\right)_{avg} \tag{3.38}$$

and

$$J_i^L = c_t^L k_i^L \left(x_i^I - x_i^L\right)_{avg} \tag{3.39}$$

where c_t^P is the total molar concentration, k_i^P is the mass transfer coefficient for a binary mixture based on a mole-fraction driving force defined in the section below, and the last terms in equations (3.38) and (3.39) are the mean mole fraction driving forces over the stage. The positive direction of mass transfer is assumed to be from the vapor phase to the liquid phase. From the definition of the molar diffusive flux:

$$\sum_{i=1}^{C} J_i = 0 \tag{3.40}$$

Thus, for the binary system (1, 2), $J_1 = -J_2$.

The general multicomponent case for mass transfer is considerably more complex than the binary case because of component-coupling effects. For example, for the ternary system (1, 2, 3), the fluxes for the first two components are:

$$J_1^V = c_t^V \kappa_{11}^V \left(y_1^V - y_1^I\right)_{avg} + c_t^V \kappa_{12}^V \left(y_2^V - y_2^I\right)_{avg} \tag{3.41}$$

$$J_2^V = c_t^V \kappa_{21}^V \left(y_1^V - y_1^I\right)_{avg} + c_t^V \kappa_{22}^V \left(y_2^V - y_2^I\right)_{avg} \tag{3.42}$$

The flux for the third component is not independent of the other two, but is obtained from equation (3-40):

$$J_3^V = -J_1^V - J_2^V \tag{3.43}$$

In these equations, the binary-pair coefficients, κ^P, are complex functions related to inverse rate functions described below and called Maxwell-Stefan mass-transfer coefficients in binary mixtures.

For the general multicomponent system (1, 2,..., C), the independent fluxes for the first C - 1 components are given in matrix equation form as

$$\mathbf{J}^V = c_t^V \left[\kappa^V\right]\left(\mathbf{y}^V - \mathbf{y}^I\right)_{avg} \tag{3.44}$$

$$\mathbf{J}^L = c_t^L \left[\kappa^L\right]\left(\mathbf{x}^I - \mathbf{x}^L\right)_{avg} \tag{3.45}$$

where $\mathbf{J}^P$, $\left(\mathbf{y}^V - \mathbf{y}^I\right)_{avg}$, and $\left(\mathbf{x}^I - \mathbf{x}^L\right)_{avg}$ are column vectors of length C - 1 and $\left[\kappa^P\right]$ is a (C – 1) x (C – 1) square matrix. The method for determining the average mole fraction driving forces depends upon the flow patterns of the vapor and liquid phases.

3.2.2 Multicomponent mass transfer coefficient (Seader & Henley, 1998)

According to the theory of Krishna and Standart (1979), the mass transfer coefficients for multicomponent systems can be calculated directly from the binary mass transfer coefficients as follows:

For ideal gas,

$$\left[\kappa_{ik}^{V}\right]a = \left[B_{ik}^{V}\right]^{-1}a \tag{3.46}$$

where the elements of B_{ik}^{V} in terms of general mole fractions, y_i, are:

$$B_{ik}^{V} = \frac{y_i}{k_{iC}^{V}} + \sum_{\substack{m=1\\ m\neq i}}^{C} \frac{y_m}{k_{im}^{V}}, \quad i = k = 1,2,\ldots,C-1 \tag{3.47}$$

and

$$B_{ik}^{V} = -y_i\left(\frac{1}{k_{ik}^{V}} - \frac{1}{k_{iC}^{V}}\right), \quad i \neq k = 1,2,\ldots,C-1 \tag{3.48}$$

For a nonideal liquid solution:

$$\left[\kappa_{ik}^{L}\right]a = \left[B_{ik}^{L}\right]^{-1}a\left[\Gamma_{ik}^{L}\right] \tag{3.49}$$

where the elements of B_{ik}^{L} in terms of general mole fractions, x_i, are:

$$B_{ik}^{L} = \frac{x_i}{k_{iC}^{L}} + \sum_{\substack{m=1\\ m\neq i}}^{C} \frac{x_m}{k_{im}^{L}}, \quad i = k = 1,2,\ldots,C-1 \tag{3.50}$$

$$B_{ik}^{L} = -x_i\left(\frac{1}{k_{ik}^{L}} - \frac{1}{k_{iC}^{L}}\right), \quad i \neq k = 1,2,\ldots,C-1 \tag{3.51}$$

and

$$\Gamma_{ik}^{L} = \delta_{ik} + x_i\left(\frac{\partial \ln \gamma_i^{L}}{\partial x_k}\right)_{T,P,x_j,j\neq k=1,\ldots,C-1} \tag{3.52}$$

For a nonideal vapor, a $\left[\Gamma^{V}\right]$ term can be included in (3.46), but this is rarely necessary. For either phase, if an equation-of-state model is used, (3.52) can be rewritten by substituting $\bar{\phi}_i$, the mixture fugacity coefficient, for γ_i. The term δ_{ij} is the Kronecker delta, which is 1 if $i = j$ and 0 if not. The thermodynamic factor is required because it is generally accepted that the fundamental driving force for diffusion in the gradient of the chemical potential rather than the mole fraction or concentration gradient.

When mass-transfer fluxes are moderate to high, an additional correlation term is needed in (3.46) and (3.49) to correct for distortion of the composition profiles.

3.2.3 Binary mass transfer coefficients (Krishnamurthy & Taylor, 1985c)

The binary mass transfer coefficients for bubble cap trays which are used to calculate multicomponent mass transfer coefficients in the section above can be predicted from the empirical correlations of the AIChE method (AIChE, 1958).

For the vapor phase,

$$k_{ik}^{V}a = \frac{\left(0.776 + 4.567W_x - 0.2377F_s + 87.319L_F\right)V}{\left[\mu_i^{V} / \left(\rho_M^{V} D_{ik}^{V}\right)\right]^{0.5}} \qquad (3.53)$$

For the liquid phase

$$k_{ik}^{L}a = 20308 {D_{ik}^{L}}^{0.5}\left(0.21313F_s + 0.15\right)t_L L_F \qquad (3.54)$$

where

$$t_L = 1.2002\, Z_C Z_L / L_F \qquad (3.55)$$

$$Z_C = 0.0254\left(1.65 + 7.4803\, W_F + 80.478\, L_F - 0.5328 F_s\right) \qquad (3.56)$$

3.2.4 **Diffusivity** (Reid et al., 1987)

Because the great accuracy and ease of use of the empirical equation of Fuller, Schetter, and Giddings, it is chosen for predicting the binary gas diffusivity:

$$D_{AB} = D_{BA} = \frac{0.00143\,T^{1.75}}{PM_{AB}^{1/2}\left[\left(\sum_{v}\right)_{A}^{1/3} + \left(\sum_{v}\right)_{B}^{1/3}\right]^{2}} \tag{3.57}$$

where D_{AB} is in cm^2/s, P is in atm, and T is in K,

$$M_{AB} = \frac{2}{(1/M_A) + (1/M_B)} \tag{3.58}$$

and $\sum_{v}$ = summation of atomic and structure diffusion volumes.

The empirical Wilke-Chang equation is used to predict liquid diffusivity:

$$D_{AB} = \frac{7.4 \times 10^{-8} (\phi_B M_B)^{1/2} T}{\mu_B v_A^{0.6}} \tag{3.59}$$

where the units are cm^2/s for D_{AB}; cP for the solvent viscosity, μ_B; K for T; and cm^3/mol for v_A, the liquid molar volume of the solute at its normal boiling point. The parameter ϕ_B is an association factor for the solvent.

3.3 Thermodynamic properties

3.3.1 **Heat transfer coefficient** (Seader & Henley, 1998)

Heat transfer coefficients for the vapor-phase film are estimated from the Chilton-Colburn analogy between heat and mass transfer:

$$h^{V} = k_{av,j}^{V} C_{pM}^{V} (Le)^{2/3} \tag{3.60}$$

where

$$Le = \frac{Sc}{Pr} \tag{3.61}$$

and heat transfer coefficients for the liquid-phase film can be estimated by using a penetration model:

$$h^L = k^L_{av,j} C^L_{pM} (Le)^{1/2} \qquad (3.62)$$

3.3.2 Enthalpy (Holland, 1981)

Enthalpy of liquid and vapor in each tray can be estimated by equation below:

$$H^V = \sum_{i=1}^{C} y_i \tilde{H}_i^V \qquad (3.63)$$

and

$$H^L = \sum_{i=1}^{C} x_i \tilde{H}_i^L \qquad (3.64)$$

where $\tilde{H}_i^V$ and $\tilde{H}_i^L$ are partial molar enthalpies of each component in vapor and liquid phase, respectively.

Partial molar enthalpy of each component can be calculated from the following equation.

For vapor, assume that it has a property like perfect gas.

$$\tilde{H}_i^V = \tilde{H}_i^\circ = \Delta H^\circ_{298,i} + \int_{298}^{T^V} C_{Pi} dT \qquad (3.65)$$

where

$$C_{pi} = a_i + b_i T + c_i T^2 + d_i T^3 \qquad (3.66)$$

For liquid,

$$\tilde{H}_i^L = \tilde{H}_i^V - \lambda_i \qquad (3.67)$$

where λ_i, latent heat of vaporization can be estimated from

$$\lambda_i = \lambda_{i,bp}\left(\frac{1-T_r}{1-T_{r,bp}}\right) \; ; \quad T_r = \frac{T}{T_C} \qquad (3.68)$$

3.3.3 Vapor-liquid equilibrium relation (Seader & Henley, 1998)

In this research, vapor-liquid equilibrium data are calculated based on modified Raoult's law which regularly being used for nonideal liquid mixtures near ambient pressure:

$$K_i = \frac{\gamma_i^L P_i^{sat}}{P} \qquad (3.69)$$

Vapor pressure, P_i^{sat}, can be calculated from Antoine's equation:

$$\ln P_i^{sat} = A_i - \frac{B_i}{C_i + T} \qquad (3.70)$$

where P_i^{sat} is in atm and T is in K.

3.3.4 Activity coefficient (Reid, et al., 1987; Seader & Henley, 1998)

Activity coefficients are the important parameter affecting to the accuracy of vapor-liquid equilibrium data prediction. In this study, the UNIFAC method is used to predict them

The UNIFAC (UNIQUAC Functional-group Activity Coefficients) method is one of the group-contribution methods which are used for predicting liquid-phase activity coefficient in mixtures. In this method, the molecular activity coefficient is separated into two parts: one part, called combinatorial part, provides the contribution due to differences in molecular size and shape, and the other, called residual part, provides the contribution due to molecular interactions.

In a multicomponent mixture, the UNIFAC equation for the activity coefficient of component i is

$$\ln\gamma_i = \ln\gamma_i^C + \ln\gamma_i^R \qquad (3.71)$$

The first group on the right-hand side accounts for combinatorial effects due to differences in molecule size and shape while the last term provides a residue contribution due to differences in intermolecular forces

An expression of an activity coefficient in the combinatorial part is

$$\ln\gamma_i^C = \ln\frac{\psi_i}{x_i} + \frac{\overline{Z}}{2} q_i \ln\frac{\theta_i}{\psi_i} + l_i - \frac{\psi_i}{x_i}\sum_{j=1}^{C} x_j l_j \qquad (3.72)$$

where x_i is the mole fraction of component i in the mixture, ψ_i is a segment fraction and θ_i is an area fraction which are defined as:

$$\psi_i = \frac{x_i r_i}{\sum_{i=1}^{C} x_i r_i}, \qquad (3.73)$$

$$\theta_i = \frac{x_i q_i}{\sum_{i=1}^{C} x_i q_i} \qquad (3.74)$$

and l_i is defined as

$$l_i = \left(\frac{\overline{Z}}{2}\right)(r_i - q_i) - (r_i - 1), \quad \overline{Z} = 10 \qquad (3.75)$$

Parameter r_i and q_i in (3.73)-(3.75) are obtained by

$$r_i = \sum_k \nu_k^{(i)} R_k, \qquad (3.76)$$

$$q_i = \sum_k \nu_k^{(i)} Q_k \qquad (3.77)$$

where $\nu_k^{(i)}$ is the number of functional groups of type k in molecule i, and R_k and Q_k are the volume and area parameters, respectively, for the type-k functional group.

The residual term in (3.71) is replaced by the expression:

$$\ln \gamma_i^R = \sum_k \nu_k^{(i)} \left(\ln \Gamma_k - \ln \Gamma_k^{(i)} \right) \tag{3.78}$$

where Γ_k is the residual activity coefficient of the functional group k in the actual mixture, and $\Gamma_k^{(i)}$ is the same quantity but in a reference mixture that contains only molecules of type i. The latter quantity is required so that $\gamma_i^R \rightarrow 1.0$ as $x_i \rightarrow 1.0$. Both Γ_k and $\Gamma_k^{(i)}$ are found from the following expression:

$$\ln \Gamma_k = Q_k \left[1 - \ln \left(\sum_m \theta_m T_{mk} \right) - \sum_m \frac{\theta_m T_{mk}}{\sum_n \theta_n T_{nm}} \right] \tag{3.79}$$

where θ_m is the area function of group m, given by an equation similar to equation (3.74),

$$\theta_m = \frac{X_m Q_m}{\sum_n X_n Q_m} \tag{3.80}$$

where X_m is the mole fraction of group m in the mixture,

$$X_m = \frac{\sum_j \nu_m^{(j)} x_j}{\sum_j \sum_n \left(\nu_n^{(j)} x_j \right)} \tag{3.81}$$

and T_{mk} is a group interaction parameter given by an equation below,

$$T_{mk} = \exp\left(-\frac{a_{mk}}{T} \right) \tag{3.82}$$

where $a_{mk} \neq a_{km}$. When $m = k$, then $a_{mk} = 0$ and $T_{mk} = 1.0$. For $\Gamma_k^{(i)}$, equation (3.79) also applies, where θ term correspond to the pure component i. Parameter a_{mk} and a_{km} are obtained from a database using a wide range of experimental results. Tables of recommended values of UNIFAC parameters R, Q, a_{mk} and a_{km} are updated periodically. These data and a detailed example of the estimation of activity coefficients by the UNIFAC method is given by Reid et al.

3.4 Solution method (Reklaitis et al., 1983)

Because the model equations are strongly nonlinear and there are many variables which are mostly limited by their constraints have to be solved, the method of minimization sum of square error is chosen for solving this large set of nonlinear equations in order to avoid convergence problems. In this work, the complex method is applied for solving a problem.

The complex method is one of direct-search methods which is modified from the simplex direct-search method by Box.

In this method, the set of trial points is generated randomly and sequentially. Given variable upper and lower bounds $x^{(U)}$ and $x^{(L)}$, the pseudo-random variable uniformly distributed on the interval (0,1) is sampled and the point coordinates calculated using equation (3.83):

$$x_i = x_i^{(L)} + R_i\left(x_i^{(U)} - x_i^{(L)}\right); \quad i = 1,2,\ldots,M \tag{3.83}$$

where R_i is a random number distributed uniformly on the interval (0,1).and x is an n-component vector.

M samples are required to define a point in M dimensions. Each newly generated point is tested for feasibility, and if infeasible, it is retracted toward the centroid of the previously generated points until it becomes feasible. The total number of points to be used, Q, should be no less than M+1 but can be larger. Given this set of points, the objective function is evaluated at each point, and the point corresponding to the highest value is rejected. A new point is generated by reflecting the rejected point a certain distance through the centroid of the remaining points. Thus, if x^R is the rejected point and $\bar{x}$ is the centroid of the remaining points, then the new point is calculated via:

$$x^m = \bar{x} + \alpha(\bar{x} - x^R) \tag{3.84}$$

The parameter α determines the distance of the reflection: $\alpha = 1$ corresponds to setting the distance $\| x^m - \bar{x} \|$ equal to $\| \bar{x} - x^R \|$; $\alpha > 1$ corresponds to an expansion and $\alpha < 1$ to a contraction.

Now, at the new point, the performance function and the constraints are evaluated. These are several alternatives:

1. The new point is feasible and its function value is not the highest of the set of points. In this case, select the point that does correspond to the highest and continue with a reflection.
2. The new point is feasible and its function value is the highest of the current set of Q points. Rather than reflecting back again, retract the point by half the distance to the previously calculated centroid and do it again and again until this function value is not the highest one.
3. The new point is infeasible. Retract the point by half the distance to the previous calculated centroid.

The search is terminated when the pattern of points has shrunk so that the points are sufficiently close together and/or when difference between the function values at the points become small enough.

Box performed numerical experiments with this algorithm and on this empirical basis recommends using $\alpha = 1.3$ and Q = 2M. Biles, on the other hand, reported good results with Q = M+2. Most implementations follow Box's recommendation that if a point exceeds one of the bounds, its corresponding coordinate be set equal to the bound. The $\alpha > 1$ compensates for the shrinking of the complex caused by halving the distances, while the large number of vertices is intended to prevent the complex from collapsing and flattening out along a constraints. The flowchart in figure 3.4a and 3.4b show the algorithm of this method.

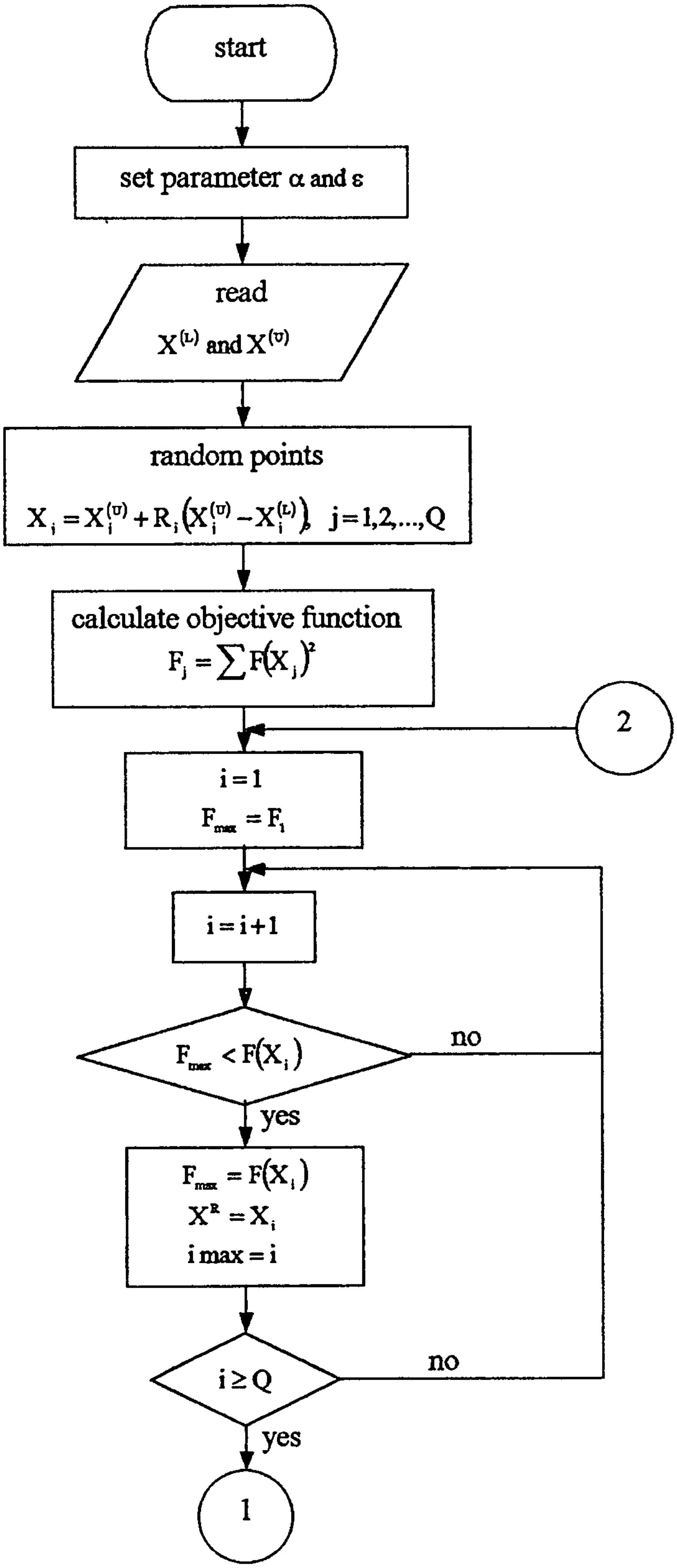

Figure 3.4a Complex algorithm flowchart for solving the set of model equations

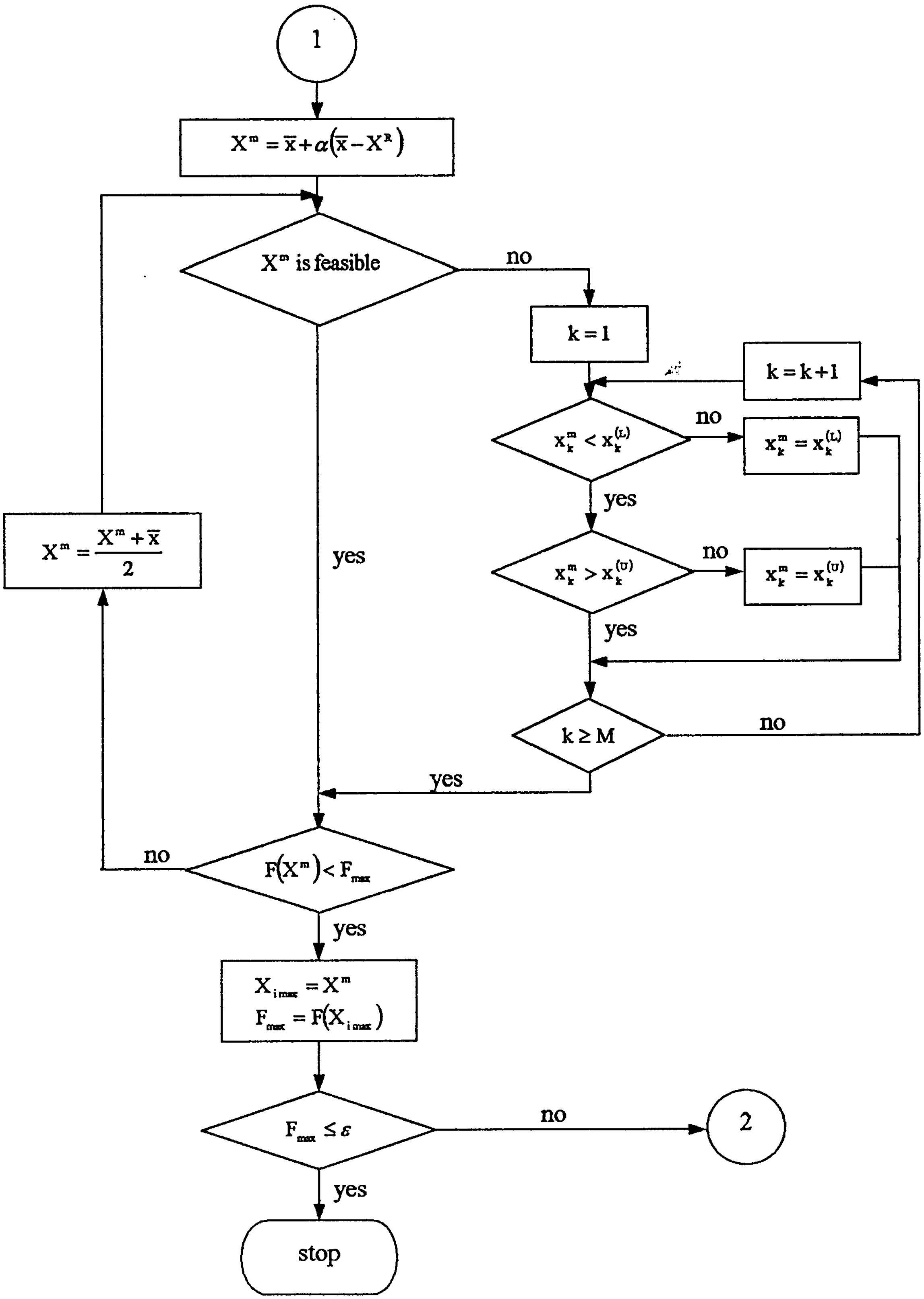

Figure 3.4b Complex algorithm flowchart for solving the set of model equations (continued)

CHAPTER 4

RESULTS AND DISCUSSION

Last chapter, it described the mathematical models to predict behaviors that occur in a reactive distillation column based on equilibrium and nonequilibrium-stage assumptions along with an algorithm. A computer program based on FORTRAN 90 language for solving the reactive distillation problem was written. In this chapter, the obtained results from the nonequilibrium model are compared with the results from the experimental data and those of the equilibrium model to prove the hypothesis that the nonequilibrium model may give better results. Therefore, the results and discussion would be explained into two parts as following.

4.1 Equilibrium model versus nonequilibrium model (both apply UNIFAC prediction)

Figure 4.1 and 4.2 show the profiles of mole fraction in the bulk liquid and vapor on each tray in the reactive distillation column based on the equilibrium and nonequilibrium models, respectively. The mole fraction profile of each species in both phases for both models is definitely different, especially for the ethanol. The mole fraction of ethanol in both phases for the nonequilibrium model is about 20% higher, except on the first tray. The assumption of equilibrium between liquid and vapor phases may cause lower ethanol concentration. Further, the equilibrium model predicts two times higher ethyl acetate yield at the top of the column than the nonequilibrium one. On the other hand, the nonequilibrium model predictics triple ethyl acetate yield at the bottom comparing with the equilibrium model.

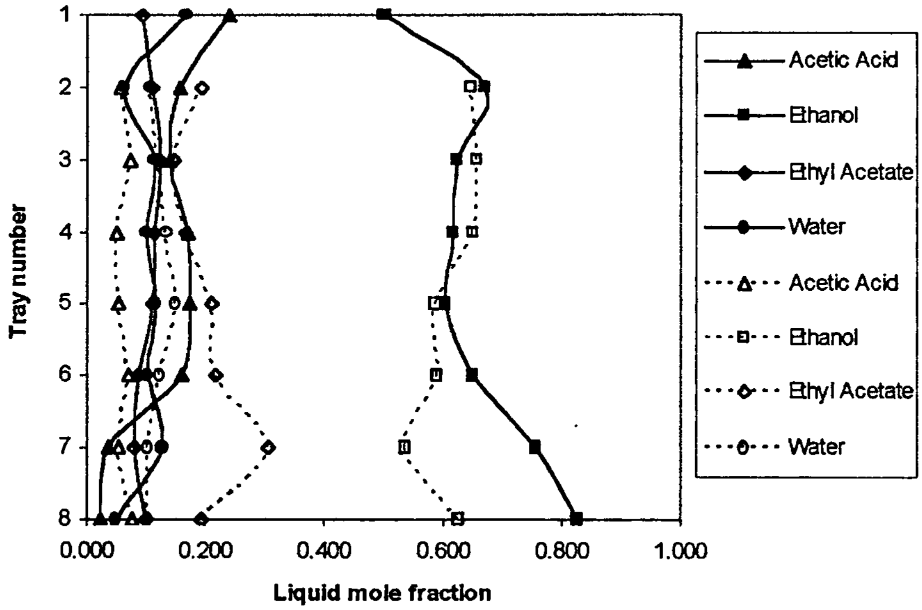

Figure 4.1 Mole fraction of liquid (——) and vapor (- - -) in the bulks of the equilibrium model.

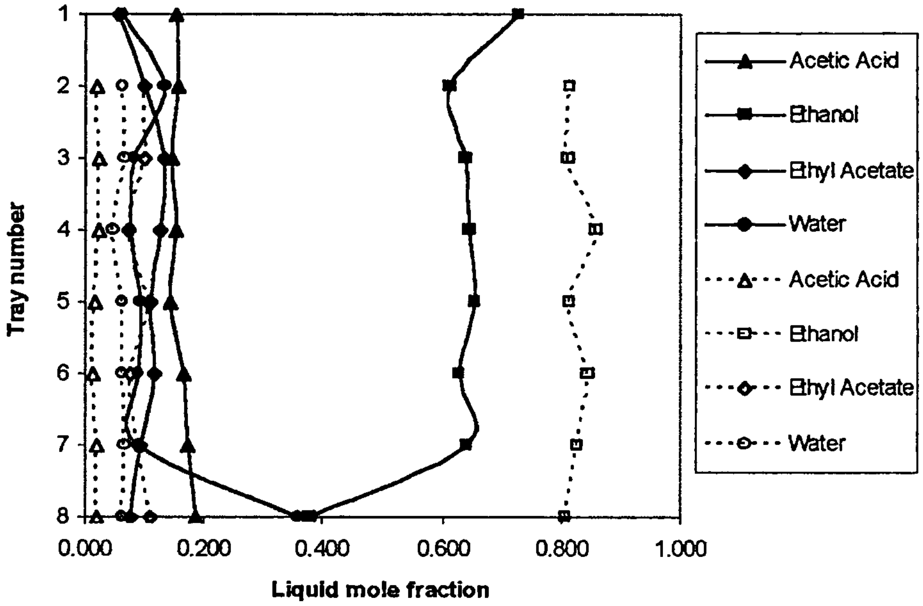

Figure 4.2 Mole fraction of liquid (——) and vapor (- - -) in the bulks of the nonequilibrium model.

Figure 4.3 and 4.4 illustrate liquid and vapor mole fraction profiles in the bulk comparing with those at the interface for the nonequilibrium model, respectively.

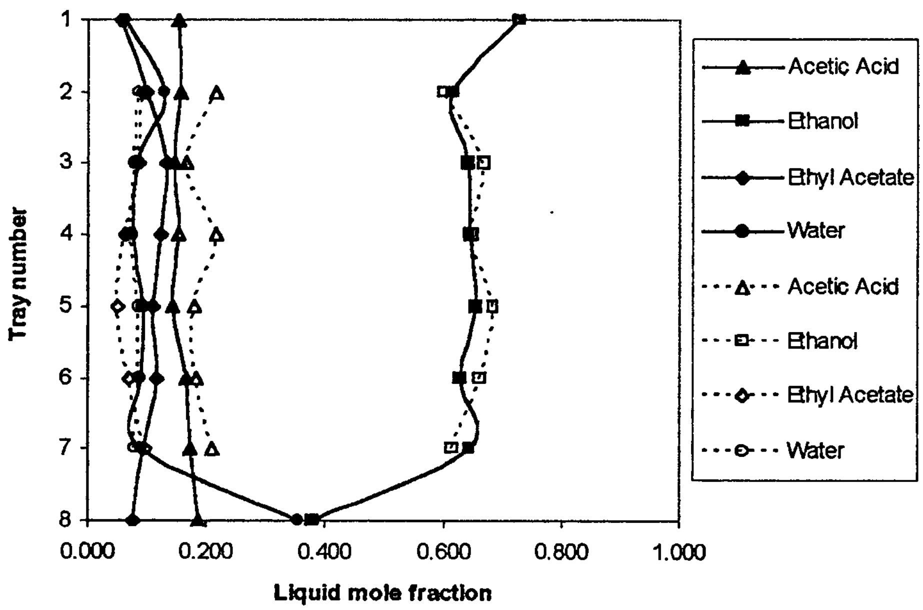

Figure 4.3 Liquid mole fraction in the bulk (——) and at the interface (- - -).

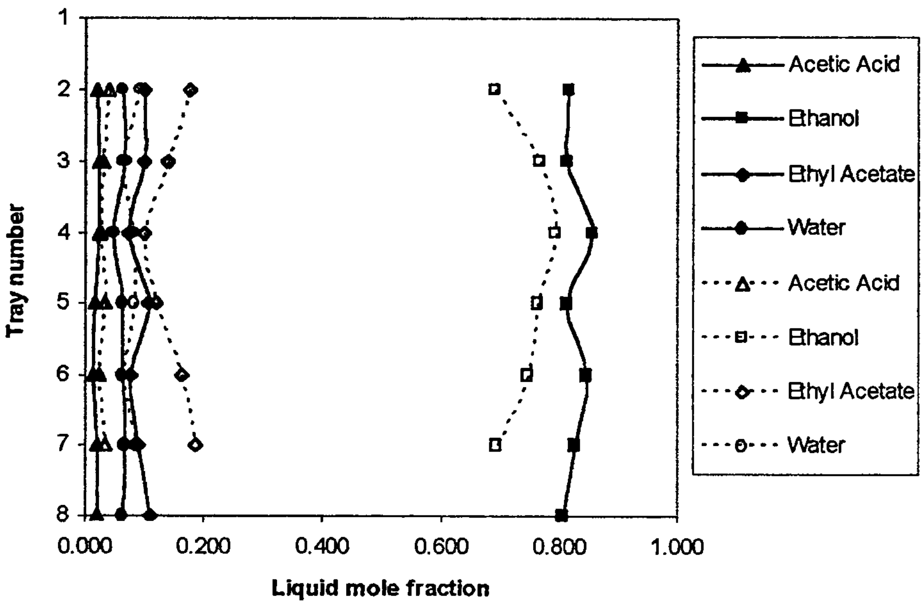

Figure 4.4 Vapor mole fraction in the bulk (——) and at the interface (- - -).

Mostly, the mole fraction profile of each species shows a transfer from one bulk fluid to the interface and from the interface to another bulk fluid following the profile as in figure 3.3. For example, ethyl acetate is always being transferred from the liquid phase to the vapor phase in this column.

As comparing to ethyl acetate, mass transfer of ethanol occurs in the opposite direction of ethyl acetate while there is a very small net transfer of water between the two phases.

In the case of acetic acid, the calculated results show that mass both in the liquid and the vapor bulk transfers to the interface. This situation cannot occur in real. It may be noted that the concentration difference between the bulk vapor and the interface is calculated to be quite small.

According to the nonequilibrium assumption, the mole fraction of each phase is in equilibrium at the interface. Figure 4.5 shows that ethanol and ethyl acetate vapor mole fractions are more than the liquid ones which agree with the volatility of these two compounds comparing with the others. But for acetic acid, mole fraction of liquid are more than that of vapor while for water mole fraction of liquid and vapor are almost equal.

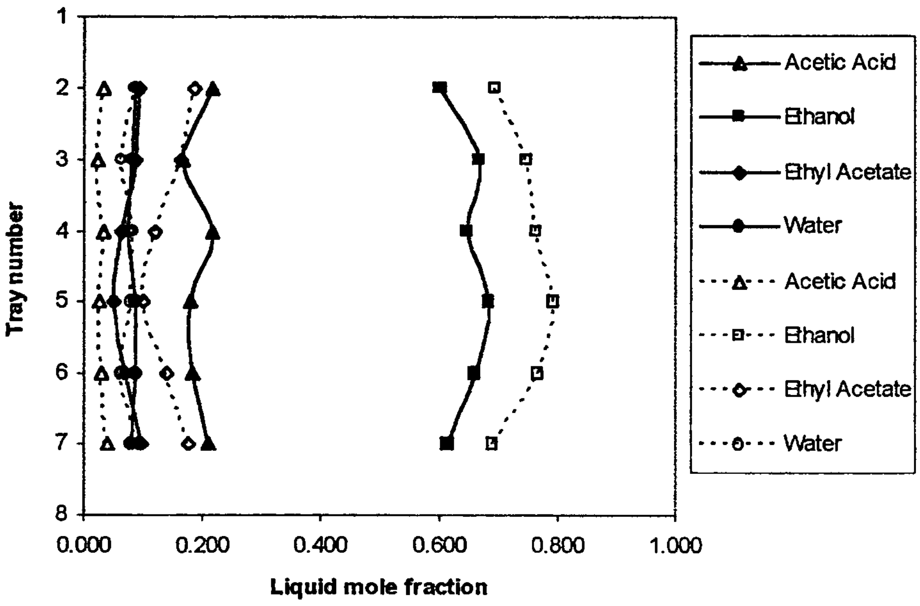

Figure 4.5 Mole fraction of liquid (——) and vapor (- - -) at the interface.

4.2 Simulation results versus experimental results

In this part, simulation results will be compared with the experimental results from the work of Komatsu (1977). Simulation results from three models will be explained shortly. They are results from :

a. equilibrium model in chapter 3

b. equilibrium model of Alejski et al. (1988) work

c. nonequilibrium model in chapter 3.

Figure 4.6-4.8 demonstrate liquid mole fraction profiles obtained from the calculation comparing with the experimental results.

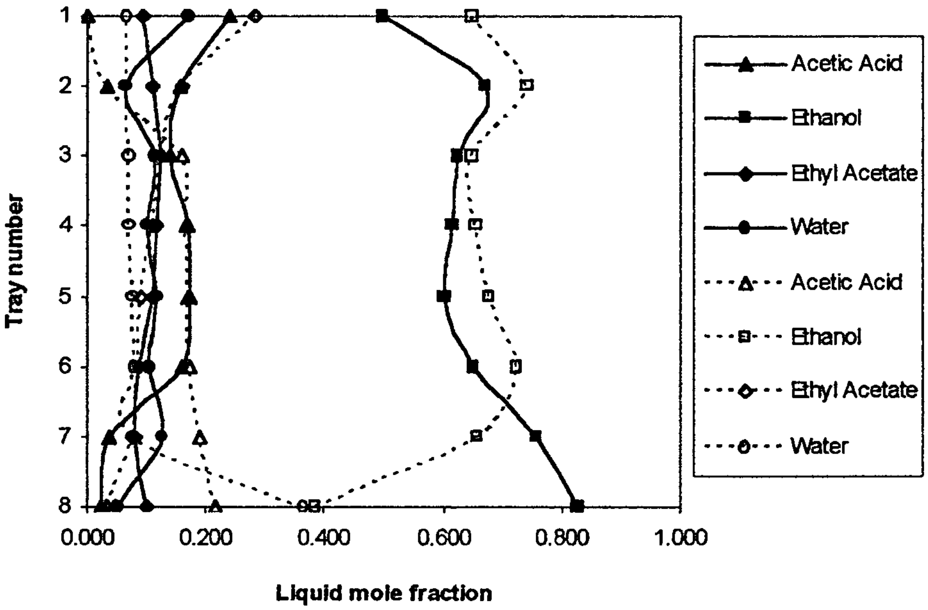

Figure 4.6 Liquid mole fraction of calculated results (——) of the equilibrium model in chapter 3 and experimental results (- - -).

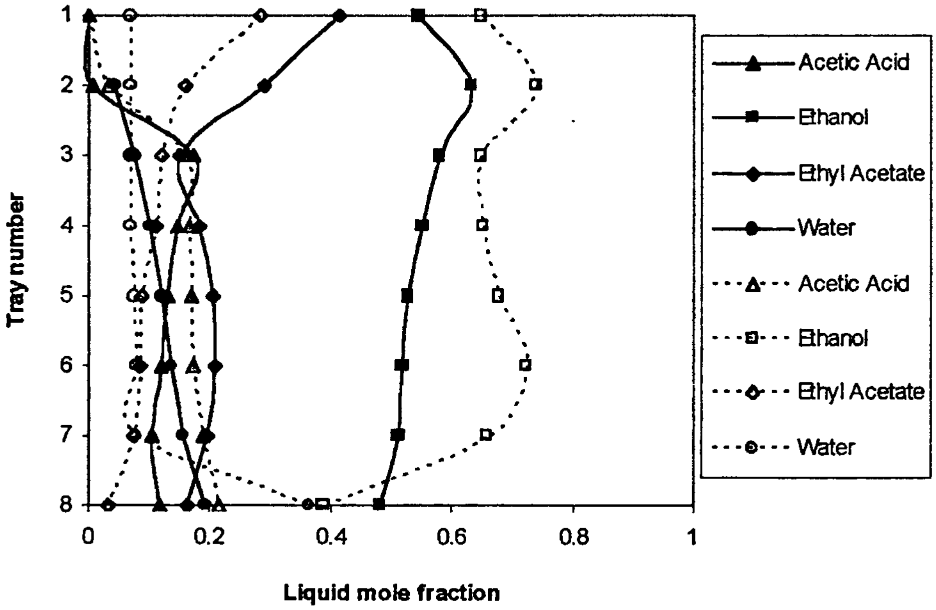

Figure 4.7 Liquid mole fraction of the results of Alejski et al., 1988 (——) and experimental results (- - -).

The simulated results in figure 4.7 proposed by Alejski et al. (1988) are obtained from the equilibrium model like those in figure 4.6 but the former work used the empirical formulas proposed in the work of Komatsu (1977) to predict vapor-liquid equilibrium constants while the UNIFAC method is used in the later one. From figure 4.6, the computed results agree satisfactorily with the experimental results, especially the compositions from tray 3 to tray 7. Comparing with the results of Alejski et al (1988) in figure 4.7, the simulation in this work show overall superior results.

In figure 4.8, the computed results are from a nonequilibrium model which vapor-liquid equilibrium constants are predicted by using the UNIFAC method. From the experimental results, when comparing these simulated results with the equilibrium model results shown in figure 4.6 and 4.7, the nonequilibrium model definitely provides much better prediction.

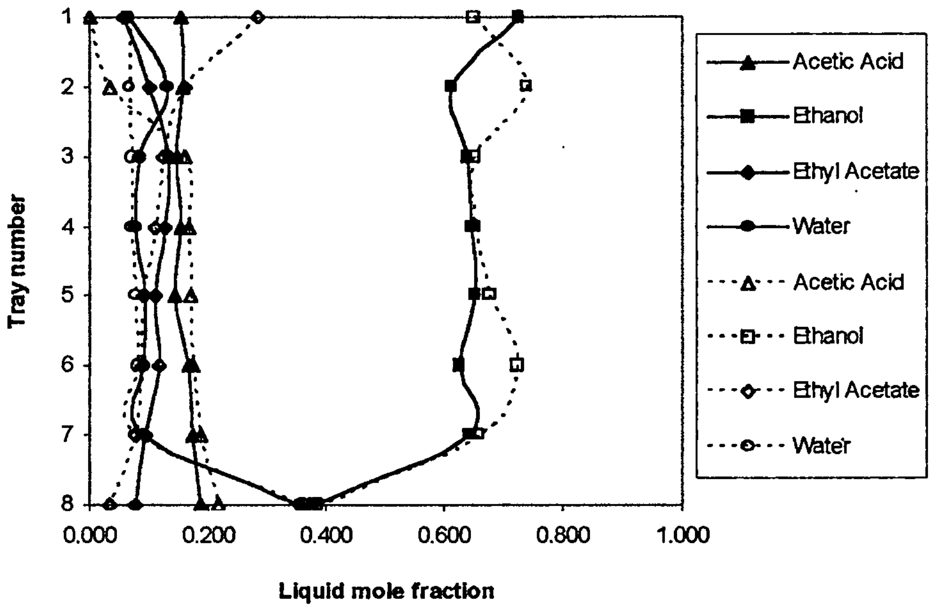

Figure 4.8 Liquid mole fraction of calculated results (——) of the nonequilibrium model and experimental results (- - -).

Considering temperature profiles, figure 4.9-4.11 display the temperature profile of liquid obtained from the simulation results of Alejski et al (1988), the equilibrium and nonequilibrium models described in chapter 3 comparing with the experimental results, respectively. Obviously, the temperature results are much better predicted by the nonequilibrium model when comparing with the equilibrium results in this work and in the work of Alejski et al. (1988).

Comparing the results in figure 4.9 with those in figure 4.10, the temperature results in tray 1 and 2 from the latter model is closer to the experimental data than the former one but in the other trays the former model results seem to be closer.

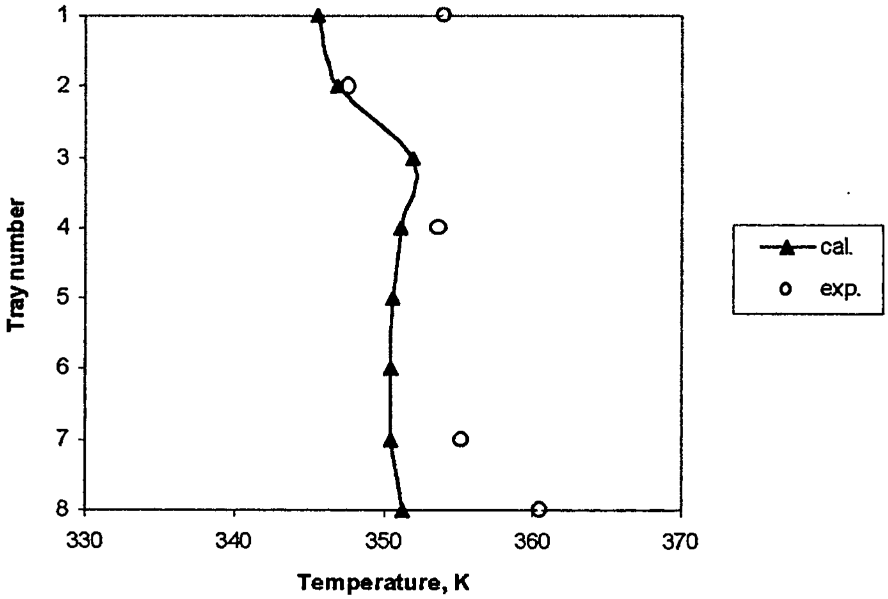

Figure 4.9 Liquid temperature of the results of Alejski et al., 1988 and experimental results.

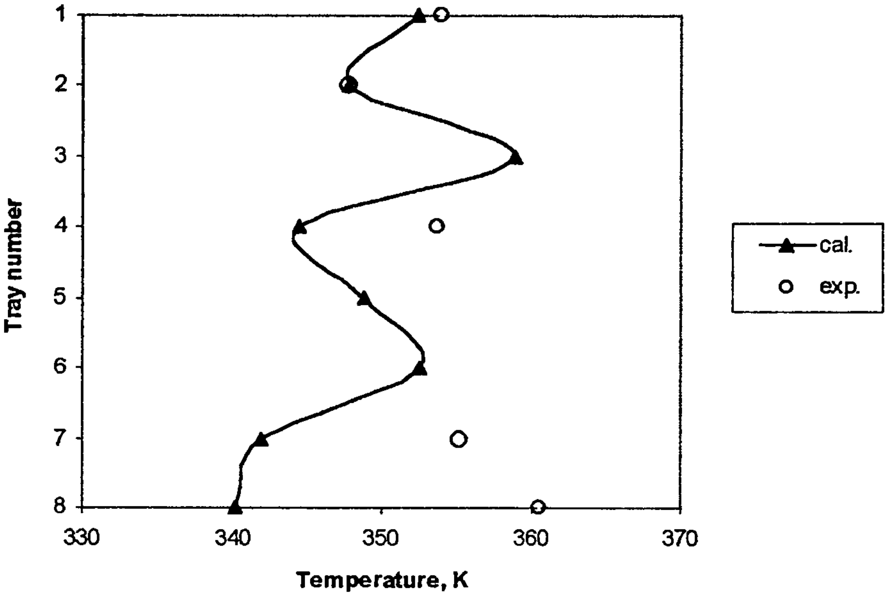

Figure 4.10 Liquid temperature of calculated results of the equilibrium model and experimental results.

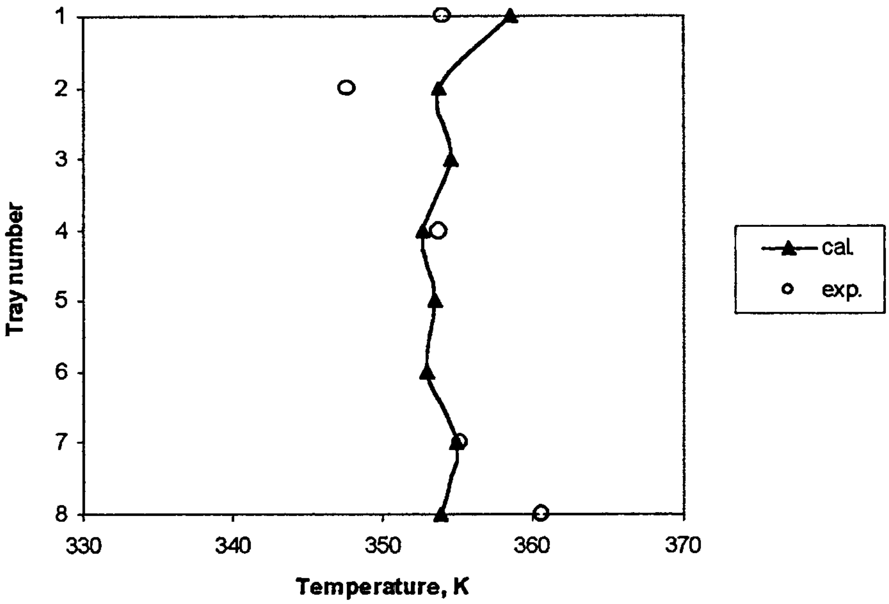

Figure 4.11 Liquid temperature of calculated results of the nonequilibrium model and experimental results.

Figure 4.12 shows the bulk liquid, bulk vapor and interface temperatures simulated from the nonequilibrium model. It can see that in each tray the temperature in the bulk liquid is a little bit lower comparing with that in the bulk vapor while the temperature at the interface is between the temperature of bulk liquid and bulk vapor. These results are similar to the pattern of temperature as seen in figure 3.3.

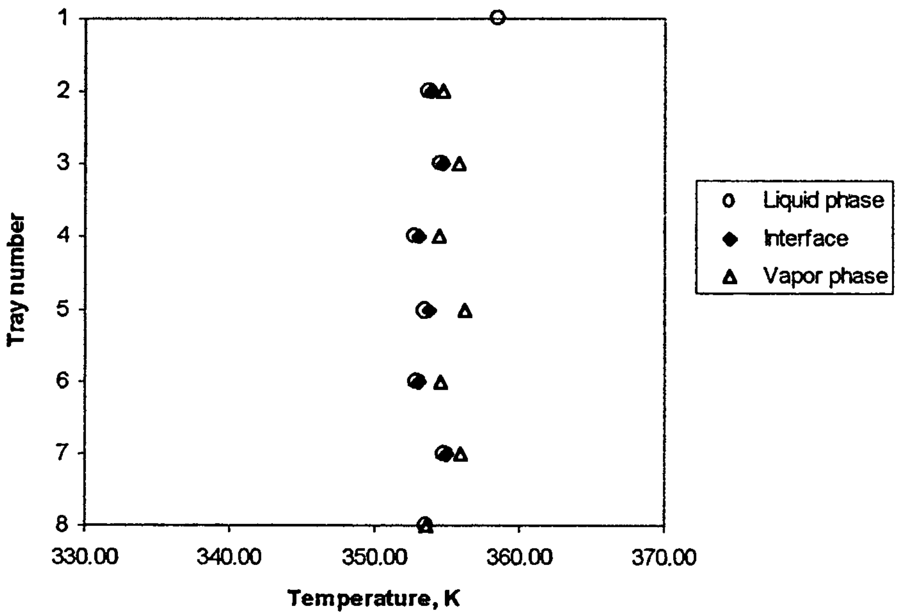

Figure 4.12 Temperature at the vapor bulk, the liquid bulk and the interface.

CHAPTER 5

CONCLUSION AND SUGGESTION

To prove the hypothesis that UNIFAC is the better method for predicting vapor-liquid equilibrium constants in the nonideal system comparing with the model results reported by Alejski et al., 1988 (which used empirical formulas proposed in the work of Komatsu, 1977, this empirical formulas has no clear original citation), the equilibrium-based model is set initially. The complex method is used to solve the model equations of this work whereas in the comparing paper, the Powell method of minimization was used. Therefore, there are two distinctive reasons that make the model results with UNIFAC prediction to be better; one could be from the method for predicting thermodynamic correlations and the other could be the applied numerical methods. Since, Okur and Bayramoglu, 2001 confirmed that the UNIFAC method provides better results of thermodynamic prediction of ethyl acetate synthesis in the reactive distillation system. UNIFAC prediction would definitely have some degree of enhancing the simulation results of this study.

The other part of this work is to compare the simulated results of the nonequilibrium-based model with those of the equilibrium-based model. In this case, both of these models applied the UNIFAC model for predicting vapor-liquid equilibrium constants and used the complex method for solve the set of model equations. The obtained results show that the nonequilibrium model provides the better results. It is undoubted that the nonequilibrium model is the superior model for predicting behaviors in a reactive distillation column, since there is no real vapor-liquid equilibrium on each tray. Although the nonequilibrium would give better results but there are some results that do not correspond to the theory, e.g. temperature in the vapor bulk, the liquid bulk and at the interface in some trays and mole fractions of vapor and liquid in the bulk and at the interface.

The unaccepted value of temperature may be from the error of the method for predicting heat transfer coefficient in multicomponent system. In the case of the missing of mole fraction, It could be possible that the error may occur from the tray sizing parameters estimation and the mass transfer coefficient prediction.

It is proposed that better expermental results may be obtained from the use of larger diameter column for doing the experiment, because too small column may cause the error of estimating tray sizing parameters.

BIBLIOGRAPHY

Abufares, A.A. and Douglas, P.L. 1995 "Mathematical Modeling and Simulation of an MTBE Catalytic Distillation Process Using SpeedUp and Aspen Plus." **Trans. IChemE. Part A.** 73: 3-12.

Agreda, V.H. et al. 1990 "High-Purity Methyl Acetate via Reactive Distillation." **Chem. Eng. Prog.** 86: 40-46.

Alejski, K. et al. 1988 "The Application of a Minimization Method for Solving Reacting-Distillation Problems." **Comput. Chem. Eng.** 12(8): 833-839.

Alejski, K. and Duprat, F. 1996 "Dynamic Simulation of the Multicomponent Reactive Distillation." **Chem. Eng. Sci.** 51(18): 4237-4252.

Barbosa, D. and Doherty, M.F. 1988 "The Influence of Equilibrium Chemical Reactions on Vapor-Liquid Phase Diagrams." **Chem. Eng. Sci.** 43(3): 529-540.

Barbosa, D. and Doherty, M.F. 1988 "The Simple Distillation of Homogeneous Reactive Mixtures." **Chem. Eng. Sci.** 43(3): 541-550.

Barbosa, D. and Doherty, M.F. 1988 "Design and Minimum-Refkux Calculations for Single-Feed Multicomponent Reactive Distillation Columns." **Chem. Eng. Sci.** 43(7): 1523-1537.

Bogacki, M.B. et al. 1989 "The Fast Method of the Solution of a Reacting Distillation Problem." **Comput. Chem. Eng.** 13(9): 1081-1085.

Bravo, J.L. and Pyhalahti, A. 1993 "Investigations in Catalytic Distillation Pilot Plant: Vapor/Liquid Equilibrium, Kinetics, and Mass-Transfer Issues." **Ind. Eng. Chem. Res.** 32: 2220-2225.

Chang, Y.A. and Seader, J.D. 1988 "Simulation of Continuous Reactive Distillation by a Homotopy-Continuation Method." **Comput. Chem. Eng.** 12(12): 1243-1255.

DeGarmo, J.L. et al. 1992. "Consider Reactive Distillation." **Chem. Eng. Prog.** 3: 43-50.

Duprat, F. and Gau, G. 1991 "Reactive Distillation of Pyridine Mixtures with an Organic Acid." **Canadian J. Chem. Eng.** 69: 1320-1334.

Fuchigimi, Y. 1990 "Hydrolysis of Methyl Acetate in Distillation Column Packed with Reactive Packing of Ion Exchange Resin." **J. Chem. Eng. Japan.** 23: 354-358.

Gallant, R.W. and Yaws, C.L. 1992 **Physical Properties of Hydrocarbons.** Volume 1. 2nd ed. Gulf Publishing Company.

Gallant, R.W. and Yaws, C.L. 1993 **Physical Properties of Hydrocarbons.** Volume 2. 3rd ed. Gulf Publishing Company.

Grosser, J.H. et al. 1987 "Modeling of Reactive Distillation Systems." **Ind. Eng. Chem. Res.** 26(5): 983-989.

Holland, C.D. 1981 **Fundamentals of Multicomponent Distillation.** New York : McGraw-Hill.

Jacobs, R. and Krishna, R. 1993 "Multiple Soluations in Reactive Distillation for Methyl tert-Butyl Ether Synthesis." **Ind. Eng. Chem. Res.** 32: 1706-1709.

Kerkhof, F.P.J.M. and Mak, A.N.S. 1993 "Multiple Steady States during Reactive Distillation of Methyl tert-Butyl Ether." **Ind. Eng. Chem. Res.** 32: 2767-2774.

Keyes, D.B. 1932. "Esterification Processes and Equipment." **Ind. Eng. Chem.** 24: 1096-1103.

Komatsu, H. et al. 1970 "Distillation Accompanied by Esterification of AcOH-EtOH." **Kogaku Kogaku.** 34: 45-52.

Komatsu, H. 1977 "Application of the Relaxation Method for Solving Reacting Distillation Problems." **J. Chem. Eng. Japan.** 10(3): 200-205.

Krishna, R. and Standart, G.L. 1979. "Mass and Energy Transfer in Multicomponent Systems." **Comput. Chem. Eng.** 3: 201-209.

Krishnamurthy, R. and Taylor, R. 1985a. "A Nonequilibrium Stage Model of Multicomponent Separation Processes. Part I: Model Description and Method of Solution." **AIChE J.** 31(3): 449-455.

Krishnamurthy, R. and Taylor, R. 1985b. "A Nonequilibrium Stage Model of Multicomponent Separation Processes. Part II: Comparison with Experiment." **AIChE J.** 31(3): 456-465.

Krishnamurthy, R. and Taylor, R. 1985c. "A Nonequilibrium Stage Model of Multicomponent Separation Processes. Part III: The Influence of Unequal Component-Efficiencies in Process Design Problems." **AIChE J.** 31(12): 1973-1985.

Lee, J.H. and Dudukovic, M.P. 1998. "A Comparison of the Equilibrium and Nonequilibrium Models for a Multicomponent Reactive Distillation Column." **Comput. Chem. Eng.** 23: 159-172.

Leyes, C.E. and Othmer, D.F. 1945. "Continuous Esterification of Butanol and Acetic Acid, Kinetic and Distillation Considerations." **Trans. AIChE.** 41: 157-196.

Leyes, C.E. and Othmer, D.F. 1945. "Esterification of Buthanol and Acetic Acid." **Ind. Eng. Chem.** 37: 968-977.

Murthy, A.K.S. 1984. "Simulation of Distillation Column Reactors." **Proc. Summer Computer Simulation Conf. Boston MA** 1: 630-635.

Okur, H. and Bayramoglu, M. 2001 "The Effect of the Liquid-Phase Activity Model on the Simulation of Ethyl Acetate Production by Reactive Distillation." **Ind. Eng. Chem. Res.**, 40: 3639-3646.

Reklaitis, G.V. et al. 1983 **Engineering Optimization; Methods and Applications.** New York : John Wiley and Sons.

Reid, R.C. et al. 1987. **The Properties of Gases and Liquids** 4th ed. New York : McGraw-Hill.

Sawastowski, H. and Pilavakis, P.A. 1979 "Distillation with Chemical Reaction in a Packed Column." **Inst. Chem. Eng. Symp. Ser.** 56: 49-63.

Seader, J.D..and Henley, E.J. 1998. **Separation Process Principled.** New York : John Wiley and Sons.

Simandl, J. and Svrcek, W.Y. "Extension of the Simultaneous-Solution and Inside-Outside Algorithms to Distillation with Chemical Reactions." **Comput. Chem. Eng.** 12(5): 337-348.

Sundmacher, K. et al. 1994 "Classification of Reactive Distillation Processes by Dimensionless Numbers." **Chem. Eng. Comm.** 127: 151-167.

Suzuki, I. et al. 1971 "Calculation of Multicomponent Distillation Accompanied by a Chemical Reaction." **J. Chem. Eng. Japan.** 4(1): 26-33.

Taylor, R. and Krishna, R. 2000 "Review: Modeling Reactive Distillation." **Chem. Eng. Sci.** 55: 5183-5229.

Wayburn, T.L. and Seader, J.D. 1988 "Homotopy-Continuation Methods for Computer Aided Process Design." **Comput. Chem. Eng.** 11: 7-25.

Zheng, Y. and Xu, X. 1992 "Study on Catalytic Distillation Processes. Part II: Simulation of Catalytic Distillation Processes – Quasi-Homogeneous and Rate-Based Model." Trans. IChemE. Part A. 70: 459-464.

Zheng, Y. and Xu, X. 1992 "Study on Catalytic Distillation Processes. Part I: Mass Transfer Characteristics in Catalyst Bed within the Column." Trans. IChemE. Part A. 70: 465-470.

APPENDIX A

SPECIFICATIONS AND TRAY SIZING PARAMETERS

A.1 Reaction data (Komatsu, 1977)

A stoichiometric equation of an esterification of acetic acid with ethanol is shown below:

$$CH_3COOH + C_2H_5OH \rightleftharpoons CH_3COOC_2H_5 + H_2O \qquad (A.1)$$

The reaction rate can be expressed as follows:

$$r = 4.76\times10^{-4}C_1C_2 - 1.63\times10^{-4}C_3C_4 \qquad (A.2)$$

A.2 Operation conditions (Komatsu, 1977)

- Feed location (from condenser)	3rd tray
- Reflux ratio	2.1
- Distillation flow rate	2.4 ml/min
- Feed flow rate (liquid at boiling point)	13 ml/min
- Feed composition (mole fraction)	
acetic acid	0.2559
ethanol	0.6159
ethyl acetate	0.0539
water	0.0743
- Bottom flow rate	10.6 ml/min
- Column pressure	1 atm

A.3 Tray sizing parameters (Komatsu, 1977; Krishnamurthy & Taylor, 1985b)

- Number of trays (include a condenser and a reboiler)	8
- Column diameter	140 mm
- Exit weir height	14 mm
- Liquid flow path length	70 mm

- Volumetric liquid hold up	
condenser	0.4 liter
tray	0.4 liter
reboiler	0.6 liter
- Type of trays	bubble cap
- Type of a condenser	total condenser
- Type of a reboiler	partial reboiler

APPENDIX B

PHYSICAL PROPERTIES

B.1 Molecular weight (Gallant & Yaws, 1992)

Table B.1 Molecular weight

Component	Molecular weight
Acetic acid	60.050
Ethanol	46.068
Ethyl acetate	88.106
Water	18.016

B.2 Boiling point and critical point (Gallant & Yaws, 1992)

Table B.2 Boiling point and critical point

Component	Boiling point at 1 atm (K)	Critical point (K)
Acetic acid	391.05	592.71
Ethanol	351.44	516.25
Ethyl acetate	350.21	523.30
Water	373.15	647.13

B.3 Molar density (Gallant & Yaws, 1992)

B.3.1 Vapor molar density

It is assumed that vapor behaves as an ideal gas so it can be estimated by using the following expression:

$$\rho^V = 12.1862\frac{P}{T^V} \qquad \text{(B.1)}$$

where pressure, P, is in atm, temperature, T^V, is in K, and density, ρ^V, is in mol/liter.

B.3.2 Liquid molar density

The liquid molar densities of consideration components are shown in table B.3

Table B.3 Liquid molar density

Component	Liquid molar density (mol/liter)
Acetic acid	17.4688
Ethanol	17.1269.
Ethyl acetate	10.2270
Water	55.5062

B.4 Heat capacity (Gallant & Yaws, 1992)

The equations for predicting heat capacity of vapor and liquid generated from experimental data are expressed in table B.4 and B.5:

Table B.4 Heat capacity of vapor

Component	heat capacity of vapor (kJ/kmol K)
Acetic acid	$C_p^V = -6.0053 \times 10^{-5} T^2 - 0.2042T + 9.5716$
Ethanol	$C_p^V = -4.6068 \times 10^{-5} T^2 - 0.0.1981T + 21.5419$
Ethyl acetate	$C_p^V = -8.8106 \times 10^{-5} T^2 - 0.3260T + 28.2027$
Water	$C_p^V = 3.6032 \times 10^{-6} T^2 - 7.2064 \times 10^{3} T + 31.2776$

Table B.5 Heat capacity of liquid

Component	Heat capacity of liquid (kJ/kmol K)
Acetic acid	$C_p^L = 0.2462T + 49.0392$
Ethanol	$C_p^L = 1.8427 \times 10^{-3} T^2 - 0.7509T + 157.5375$
Ethyl acetate	$C_p^L = 8.8106 \times 10^{-4} T^2 - 0.4493T + 188.7671$
Water	$C_p^L = 9.008 \times 10^{-9} T^4 - 1.4413 \times 10^{-5} T^3 + 9.008 \times 10^{-3} T^2 - 2.1097T + 280.0047$

B.5 Thermal Conductivity (Gallant & Yaws, 1992)

The expressions for estimating thermal conductivity of vapor and liquid are also generated from experimental data like those of heat capacity. The expressions are shown in table B.6 and B.7

Table B.6 Thermal conductivity of vapor

Component	Thermal conductivity of vapor (10^3 kW/m K)
Acetic acid	$k = 1.0\times10^{-7}T^2 - 8\times10^{-7}T + 0.0017$
Ethanol	$k = 7.0\times10^{-8}T^2 + 7.0\times10^{-5}T - 0.0114$
Ethyl acetate	$k = 1.0\times10^{-7}T^2 - 5.0\times10^{-6}T + 0.0028$
Water	$k = 4.0\times10^{-8}T^2 + 5.0\times10^{-5}T - 0.0007$

Table B.7 Thermal conductivity of liquid

Component	Thermal conductivity of liquid (10^3 kW/m K)
Acetic acid	$k = -0.0002T + 0.2139$
Ethanol	$k = -0.0003T + 0.2524$
Ethyl acetate	$k = -0.0004T + 0.2493$
Water	$k = -7.0\times10^{-9}T^3 - 3.0\times10^{-6}T^2 - 0.005T + 0.4726$

B.6 Vapor pressure (Holland, 1981)

Vapor pressure used to evaluate vapor-liquid equilibrium constants is calculated from Antoine's equation:

$$\log P^{sat} = A_i - \frac{B_i}{T - C_i} \qquad \text{(B.2)}$$

where P^{sat} is in mmHg and T is in $°C$. The constants A_i, B_i, and C_i are illustrated in table B.8.

Table B.8 Antoine's constants

Component	A	B	C
Acetic acid	7.18807	1416.70	211.00
Ethanol	8.04494	1554.30	222.65
Ethyl acetate	7.09808	1238.71	217.00
Water	7.96681	1668.21	228.00

B.7 Viscosity of vapor (Gallant & Yaws, 1992)

Vapor viscosity of each component used for predicting mass transfer coefficient in a vapor phase is calculated from the formula generated by using experimental data as shown in table B.9:

Table B.9 Viscosity of vapor

Component	Viscosity of vapor (cP)
Acetic acid	$\mu = -1.0\times10^{-10}T^3 + 4.0\times10^{-7}T^2 - 1.756\times10^{-4}T + 0.0023$
Ethanol	$\mu = 1.0\times10^{-10}T^3 - 2.0\times10^{-7}T^2 - 3.945\times10^{-4}T + 0.0012$
Ethyl acetate	$\mu = -3.0\times10^{-23}T^3 - 5.0\times10^{-8}T^2 - 2.822\times10^{-4}T + 0.0003$
Water	$\mu = -7.0\times10^{-9}T^2 - 4.088\times10^{-4}T + 0.0029$

APPENDIX C

COMPUTER PROGRAM

```
program Reactive_Distillation

          implicit none
          double precision, allocatable, dimension (:) :: TL, TV, TF, P, Vfr, Lfr, MV1, MV2, MV3, MV4, &
                                                          ML1, ML2, ML3, ML4, EV, EL, EI, RV1, RV2, &
                                                          RV3, RL1, RL2, RL3, QI1, QI2, QI3, QI4, SV, &
                                                          SL, fV1, fV2, fV3, fV4, fL1, fL2, fL3, fL4, N1, &
                                                          N2, N3, N4, QV, QL, TI, r1, r2, r3, r4, U, haL, &
                                                          haV, M1, M2, M3, M4, E, Q1, Q2, Q3, Q4, &
                                                          feedL, Sxl, Syl, Tlr, Tvr, Tir
          double precision, allocatable, dimension (:,:) :: l, v, xl, yl, x, y, z, K, Kl, Psat, Ind_var
          double precision :: D, QC, QR, reflux
          double precision, parameter :: Tbw = 373.
          integer :: j, nEq, np, nv
          integer, parameter :: Comp = 4, Limit = 90, n = 8

          nv = 23
          nEq =5+nv*(n-2)+9
          np = nEq

          allocate (TL(1:n), TV(1:n), TF(1:n), P(1:n), Vfr(1:n), Lfr(1:n), MV1(1:n), MV2(1:n), MV3(1:n), &
                   MV4(1:n), ML1(1:n), ML2(1:n), ML3(1:n), ML4(1:n), EV(1:n), EL(1:n), EI(1:n), &
                   RV1(1:n), RV2(1:n), RV3(1:n), RL1(1:n), RL2(1:n), RL3(1:n), QI1(1:n), QI2(1:n), &
                   QI3(1:n), QI4(1:n), SV(1:n), SL(1:n), fV1(1:n), fV2(1:n), fV3(1:n), fV4(1:n), fL1(1:n), &
                   fL2(1:n), fL3(1:n), fL4(1:n), U(1:n), N1(1:n), N2(1:n), N3(1:n), N4(1:n), r1(1:n), &
                  r2(1:n), r3(1:n), r4(1:n), QV(1:n), TI(1:n), haL(1:n), haV(1:n), M1(1:n), M2(1:n), &
                  M3(1:n), M4(1:n), E(1:n), Q1(1:n), Q2(1:n), Q3(1:n), Q4(1:n), feedL(1:n), Sxl(1:n), &
                  Syl(1:n), Tvr(1:n), Tlr(1:n), Tir(1:n))

          allocate (l(1:Comp,1:n), v(1:Comp,1:n), xl(1:Comp,1:n), yl(1:Comp,1:n), x(1:Comp,1:n),&
                   y(1:Comp,1:n),z(1:Comp,1:n),K(1:Comp,1:n),Kl(1:Comp,1:n),Psat(1:Comp,1:n))

          allocate (Ind_var(1:nEq,0:np))
```

```
do j = 1, n
        P(j) = 1.0
        if(j /= n) then
                U(j) = 0.4
        else
                U(j) = 0.6
        end if
end do

D = 0.0425/60                                           ! Distillate, mol/s
Reflux = 2.1

call Feed (n, TF, z, P, fL1, fL2, fL3, fL4, Comp, Limit)
call ComplexA (n, nEq, np, Ind_var, Tbw, Comp, Limit, D, QC, QR, Tvr, Tlr, Tir, TL, TV, TI, &
                SV, SL, fV1, fV2, fV3, fV4, fL1, fL2, fL3, fL4, U, N1, N2, N3, N4, r1, r2, r3, &
                r4, QV, QL, Vfr, Lfr, P, TF, v, l, x, y, z, xI, yI, Reflux)

do j = 1, n
        TL(j) = Tbw*Tlr(j)
        TV(j) = Tbw*Tvr(j)
        TI(j) = Tbw*Tir(j)
end do

open (12, file = 'TempLiq', status = 'old')
write (12,120) TL(1:n)
120 format (' ', 1ES17.6)
open (13, file = 'TempIntf', status = 'old')
write (13,120) TI(1:n)
open (14, file = 'TempVap', status = 'old')
write (14,120) TV(1:n)

open (15, file = 'N1', status = 'old')
write (15,120) N1(1:n)
open (16, file = 'N2', status = 'old')
write (16,120) N2(1:n)
```

```
    open (17, file = 'N3', status = 'old')
    write (17,120) N3(1:n)
    open (18, file = 'N4', status = 'old')
    write (18,120) N4(1:n)

    open (19, file = 'VapF', status = 'old')
    write (19,130) v(1:4,1:n)
    130 format (' ', 4ES17.6)
    open (20, file = 'LiqF', status = 'old')
    write (20,130) l(1:4,1:n)
    open (21, file = 'Xintf', status = 'old')
    write (21,130) xl(1:4,1:n)
    open (22, file = 'Yintf', status = 'old')
    write (22,130) yl(1:4,1:n)

end program Reactive_Distillation



Subroutine Bubble_Point (n, P, xx, Psat, K, TT, Tav, j, Comp, Limit)

    implicit none
    integer, intent (in) :: n
    double precision, dimension (1:n) :: TT, P
    double precision, dimension (1:4,1:n) :: xx, Psat, K
    double precision, dimension (1:4) :: A, B, C, ac
    double precision :: sumKx, error, T0, T1, Tav, tol
    integer :: i, j, Comp, Limit, mm

    tol = 1E-03
    T0 = 273.0
    T1 = 383.0

    do mm = 1, Limit
        Tav = (T0 + T1)/2
        TT(j) = Tav
```

```
                call UNIFAC(xx, n, TT, ac, j, Comp, Limit)

                A(1:4) = (/7.18807, 8.04494, 7.09808, 7.96681/)
                B(1:4) = (/1416.7, 1554.3, 1238.71, 1668.21/)
                C(1:4) = (/211.0, 222.65, 217.0, 228.0/)

                do i = 1, Comp
                        Psat(i,j) = (10**(A(i) - (B(i)/((Tav-273.13) + C(i)))))/760
                end do

                do i = 1, Comp
                        K(i,j) = ac(i)*Psat(i,j)/P(j)
                end do

                sumKx = 0.
                do i = 1, Comp
                        sumKx = sumKx + K(i,j)*xx(i,j)
                end do

                if (sumKx > 1.) then
                        T1 = Tav
                        T0 = T0
                else if (sumKx < 1) then
                        T0 = Tav
                        T1 = T1
                end if

                error = Abs(1-sumKx)/sumKx

                if (error < tol .or. mm > Limit) then
                        exit
                end if
        end do

end Subroutine Bubble_Point
```

```
Subroutine Combinatorial (xx, ll, mvp, marp, sf, af, lacc, frac1, frac2, frac, i, j, Comp, n)

    implicit none
    integer, intent (in) :: n
    double precision, dimension (1:4,1:n) :: xx
    double precision, dimension (1:90) :: ll, mvp, marp, sf, af, lacc
    double precision :: frac1, frac2, frac
    integer :: i, j, Comp

    do i = 1, Comp
        ll(i) = 5*(mvp(i)-marp(i)) - (mvp(i)-1)
    end do
    frac1 = 0.
    frac2 = 0.
    frac = 0.
    do i = 1, Comp
        frac1 = frac1 + xx(i,j)*mvp(i)
        frac2 = frac2 + xx(i,j)*marp(i)
        frac = frac + xx(i,j)*ll(i)
    end do
    do i = 1, Comp
        sf(i) = xx(i,j)*mvp(i)/frac1
        af(i) = xx(i,j)*marp(i)/frac2
    end do
    do i = 1, Comp
        lacc(i) = 0.
        if ( xx(i,j) /= 0.0) then
            lacc(i) = (Log(sf(i)/xx(i,j)))+(5*marp(i)*Log(af(i)/sf(i)))+ll(i)-(frac*(sf(i)/xx(i,j)))
        end if
    end do

end Subroutine Combinatorial
```

```
Subroutine ComplexA (n, nEq, np, Ind_var, Tbw, Comp, Limit, D, QC, QR, Tvr, Tlr, Tir, TL, TV, Tl, &
                    SV, SL, fV1, fV2, fV3, fV4, fL1, fL2, fL3, fL4, U, N1, N2, N3, N4, r1, r2, r3, r4, &
                    QV, QL, Vfr, Lfr, P, TF, v, l, x, y, z, xl, yl, Reflux)


     implicit none
     integer, intent (in) :: n, nEq, np
     double precision, dimension (1:nEq) :: xU, xL, xave, xm, IndV
     double precision, dimension (1:nEq,1) :: Fx
     double precision, dimension (1:nEq,0:np) :: Ind_var, r
     double precision, dimension (0:np) :: Func
     double precision :: Tbw, D, QC, QR, sum, Fmax, alpha, Fave, testF, testx, norm, weight, &
                         Fmin, Reflux, dTi, dy, dx
     integer :: i, j, jp, m, Comp, Limit, jmax, jmin, mm, ii
     double precision, dimension (1:n) :: Tvr, Tlr, Tir, TL, TV, Tl, SV, SL, fV1, fV2, fV3, fV4, fL1, &
                                         fL2, fL3, fL4, U, N1, N2, N3, N4, r1, r2, r3, r4, QV, QL, &
                                         Vfr, Lfr, P, TF
     double precision, dimension (1:4,1:n) :: v, l, x, y, z, xl, yl, dxl, dyl
     integer, parameter :: maxits = 3000, iters = 5
     double precision, parameter :: err1 = 1E-05, err2 = 1E-05


     open (10, file = 'BoundL', status = 'old')
     read (10,*) xL(1:nEq)
     open (11, file = 'BoundU', status = 'old')
     read (11,*) xU(1:nEq)


     call random_number(r)


     do jp = 0, np
          do j = 1, nEq
               Ind_var(j,jp) = xL(j) + r(j,jp)*(xU(j)-xL(j))
          end do
     end do
```

```
do jp = 0, np
          call ObjectiveF (n, nEq, jp, Ind_var, SV, SL, fV1, fV2, fV3, fV4, fL1, fL2, fL3, fL4, U, &
                              QV, QL, Vfr, Lfr, P, v, l, x, y, z, Func, Tbw, TF, D, QC, QR, Comp, &
                              Limit, np, Fx)
end do

do ii = 1, maxits
          Fmax = 0.
          do m = 0, np
                    if (Func(m) > Fmax) then
                              Fmax = Func(m)
                              jmax = m
                    end if
          end do

          do j = 1, nEq
                    sum = 0.
                    do m = 0, np
                              sum = sum + Ind_var(j,m)
                    end do
                    xave(j) = sum/(np+1)
          end do

          sum = 0.
          do m = 0, np
                    sum = sum + Func(m)
          end do
          Fave = sum/(np+1)

          sum      = 0.
          do m = 0, np
                    sum = sum + ((Func(m) - Fave)**2)
          end do
          testF = Sqrt(sum)
```

```
sum = 0.
do m = 0, np
    norm = 0.
    do j = 1, nEq
        norm = norm + (Ind_var(j,m) - xave(j))**2
    end do
    sum = sum + Sqrt(norm)
end do
testx = sum

if (Fmax < err1) exit

alpha = 1.3
do j = 1, nEq
    xm(j) = xave(j) + alpha*(xave(j) - Ind_var(j,jmax))
    Ind_var(j,jmax) = xm(j)
end do

do j = 1, nEq
    if (xm(j) < xL(j)) then
        xm(j) = xL(j)
    else if (xm(j) > xU(j)) then
        xm(j) = xU(j)
    end if
    Ind_var(j,jmax) = xm(j)
end do

A : do mm = 1, iters
    jp = jmax
    call ObjectiveF (n, nEq, jp, Ind_var, SV, SL, fV1, fV2, fV3, fV4, fL1, fL2, &
                fL3, fL4, U, QV, QL, Vfr, Lfr, P, v, l, x, y, z, Func, Tbw, TF, &
                D, QC, QR, Comp, Limit, np, Fx)

    if (Func(jmax) >= Fmax .and. mm <= iters) then
```

```
                        do j = 1, nEq
                                xm(j) = (Ind_var(j,jmax) + xave(j))/2
                                if (xm(j) < xL(j)) then
                                        xm(j) = xL(j)
                                else if (xm(j) > xU(j)) then
                                        xm(j) = xU(j)
                                end if
                                Ind_var(j,jmax) = xm(j)
                        end do
                else
                        exit A
                end if
        end do A
end do

Fmin = Fmax
do j = 0, np
        if (Func(j) < Fmin) then
                jmin = j
        end if
end do

jp = jmin
do i = 1, Comp
         l(i,1) = Ind_var(i,jp)
end do
Tlr(1) = Ind_var(5,jp)

do m = 2, n-1
        do i = 1, Comp
                v(i,m) = Ind_var(5+i+23*(m-2),jp)
        end do
        Tvr(m) = Ind_var(10+23*(m-2),jp)
        Tlr(m) = Ind_var(11+23*(m-2),jp)
```

```
            do i = 1, Comp
                  l(i,m) = lnd_var(11+i+23*(m-2),jp)
            end do


            N1(m) = lnd_var(16+23*(m-2),jp)
            N2(m) = lnd_var(17+23*(m-2),jp)
            N3(m) = lnd_var(18+23*(m-2),jp)
            N4(m) = lnd_var(19+23*(m-2),jp)


            xl(3,m) = lnd_var(20+23*(m-2),jp)
            xl(2,m) = lnd_var(21+23*(m-2),jp)
            xl(1,m) = lnd_var(22+23*(m-2),jp)
            yl(2,m) = lnd_var(23+23*(m-2),jp)
            yl(3,m) = lnd_var(24+23*(m-2),jp)
            yl(1,m) = lnd_var(25+23*(m-2),jp)
            Tir(m) = lnd_var(26+23*(m-2),jp)
            xl(4,m) = lnd_var(27+23*(m-2),jp)
            yl(4,m) = lnd_var(28+23*(m-2),jp)
      end do


      m = n
      do i = 1, Comp
            v(i,m) = lnd_var(5+23*(m-2)+i,jp)
      end do
      Tlr(m) = lnd_var(5+23*(m-2)+5,jp)
      Tvr(m) = Tlr(m)
      do i = 1, Comp
            l(i,m) = lnd_var(5+23*(m-2)+5+i,jp)
      end do

end Subroutine ComplexA
```

```
Subroutine Diffusivity (DL, DV, TL, TV, P, visL, j, n, DLave, DVave, Comp)

          implicit none
          integer, intent(in) :: n, Comp
          double precision, dimension (1:Comp) :: visL
          double precision, dimension (1:Comp,1:Comp) :: DL, DV
          double precision, dimension (1:n) :: TV, TL, P, DLave, DVave
          integer :: j

          ! 1 = acetic acid, 2 = ethanol, 3 = ethyl acetate, 4 = water
          ! Diffusivity in vapor phase (m^2/s), T(K), P(atm)

                    DV(1,1) = (0.00143*(TV(j)**1.75))/(438.78*P(j))/10000
                    DV(2,2) = (0.00143*(TV(j)**1.75))/(377.12*P(j))/10000
                    DV(3,3) = (0.00143*(TV(j)**1.75))/(777.84*P(j))/10000
                    DV(4,4) = (0.00143*(TV(j)**1.75))/(82.60*P(j))/10000
                    DV(1,2) = (0.00143*(TV(j)**1.75))/(405.02*P(j))/10000
                    DV(2,1) = DV(1,2)
                    DV(1,3) = (0.00143*(TV(j)**1.75))/(584.17*P(j))/10000
                    DV(3,1) = DV(1,3)
                    DV(1,4) = (0.00143*(TV(j)**1.75))/(187.53*P(j))/10000
                    DV(4,1) = DV(1,4)
                    DV(2,3) = (0.00143*(TV(j)**1.75))/(533.09*P(j))/10000
                    DV(3,2) = DV(2,3)
                    DV(2,4) = (0.00143*(TV(j)**1.75))/(179.12*P(j))/10000
                    DV(4,2) = DV(2,4)
                    DV(3,4) = (0.00143*(TV(j)**1.75))/(249.72*P(j))/10000
                    DV(4,3) = DV(3,4)

                    DVave(j) = 0.
                    DVave(j) = (DV(1,2)+DV(1,3)+DV(1,4)+DV(2,3)+DV(2,4)+DV(3,4))/6

          ! Diffusivity in liquid phase (m^2/s), T(K), P(atm)

                    DL(1,1) = ((7.4E-08)*7.7782*TL(j)/(12.0117*visL(1)))/10000
                    DL(1,2) = ((7.4E-08)*8.3128*TL(j)/(12.0117*visL(2)))/10000
```

```
          DL(1,3) = ((7.4E-08)*9.3862*TL(j)/(12.0117*visL(3)))/10000
          DL(1,4) = ((7.4E-08)*6.8441*TL(j)/(12.0117*visL(4)))/10000
          DL(2,1) = ((7.4E-08)*7.7782*TL(j)/(11.7640*visL(1)))/10000
          DL(2,2) = ((7.4E-08)*8.3128*TL(j)/(11.7640*visL(2)))/10000
          DL(2,3) = ((7.4E-08)*9.3862*TL(j)/(11.7640*visL(3)))/10000
          DL(2,4) = ((7.4E-08)*6.8441*TL(j)/(11.7640*visL(4)))/10000
          DL(3,1) = ((7.4E-08)*7.7782*TL(j)/(16.4991*visL(1)))/10000
          DL(3,2) = ((7.4E-08)*8.3128*TL(j)/(16.4991*visL(2)))/10000
          DL(3,3) = ((7.4E-08)*9.3862*TL(j)/(16.4991*visL(3)))/10000
          DL(3,4) = ((7.4E-08)*6.8441*TL(j)/(16.4991*visL(4)))/10000
          DL(4,1) = ((7.4E-08)*7.7782*TL(j)/(5.9176*visL(1)))/10000
          DL(4,2) = ((7.4E-08)*8.3128*TL(j)/(5.9176*visL(2)))/10000
          DL(4,3) = ((7.4E-08)*9.3862*TL(j)/(5.9176*visL(3)))/10000
          DL(4,4) = ((7.4E-08)*6.8441*TL(j)/(5.9176*visL(4)))/10000

          DLave(j) = 0.
          DLave(j) = (DL(1,2)+DL(1,3)+DL(1,4)+DL(2,3)+DL(2,4)+DL(3,4))/6

end Subroutine Diffusivity



Subroutine Enthalpies(n, HL, HV, HLF, HVF, x, y, z, TL, TV, P, TF, HPL, HPV, Comp)


     implicit none
     integer, intent (in) :: n, Comp
     double precision, dimension (1:n) :: HL, HV, HLF, HVF, TL, TV, TF, P, denV, denLm, &
                                          CpVm, CpLm, TcVm, TcLm
     double precision, dimension(1:Comp,1:n) :: HPL, HPV, HPLf, HPVf, x, y, z, CpV, CpL, &
                                                cV, TcL
     double precision, dimension(1:Comp) :: denL, visL, visV, lambda, lambdaf
     double precision, parameter :: unit = 1000000
     integer :: i, j
```

```
do j = 1, n

   ! CALCULATE PARTIAL MOLAR ENTHALPIES FOR LIQUID PHASE        (MJ/mol)

      if (j > 1) then
            HPV(1,j) = 4.1841*(-10607.89+(1.156*TV(j))+(3.0435E-02*(TV(j)**2))
                        -(1.3956E-05*(TV(j)**3))+(2.955E-09*(TV(j)**4)))/unit
            HPV(2,j) = 4.1841*(-58855.738+(2.153*TV(j))+(2.5565-02*(TV(j)**2))
                        -(6.68E-06*(TV(j)**3))+(8.2E-11*(TV(j)**4)))/unit
            HPV(3,j) = 4.1841*(-110238.9+(1.728*TV(j))+(4.8625*(TV(j)**2))
                        -(1.665E-05*(TV(j)**3))+(1.729E-09*(TV(j)**4)))/unit
            HPV(4,j) = 4.1841*(60135.845+(7.701*TV(j))+(2.2975-04*(TV(j)**2))
                        +(8.4E-07*(TV(j)**3))-(2.15E-10*(TV(j)**4)))/unit
      end if

   ! CALCULATE PARTIAL MOLAR ENTHALPIES FOR LIQUID PHASE        (MJ/mol)

         ! calculate latent heat of vaporization

            lambda(1) = 5600*(((1-(TL(j)/594.4))/0.3420)**0.38)
            lambda(2) = 9260*(((1-(TL(j)/516.2))/0.3191)**0.38)
            lambda(3) = 7700*(((1-(TL(j)/523.2))/0.3305)**0.38)
            lambda(4) = 9717*(((1-(TL(j)/647.3))/0.4235)**0.38)


            HPL(1,j) = 4.1841*(-10607.89+(1.156*TL(j))+(3.0435E-02*(TL(j)**2))
                        -(1.3956E-05*(TL(j)**3))+(2.955E-09*(TL(j)**4))-lambda(1))/unit
            HPL(2,j) = 4.1841*(-58855.738+(2.153*TL(j))+(2.5565-02*(TL(j)**2))
                        -(6.68E-06*(TL(j)**3))+(8.2E-11*(TL(j)**4))-lambda(2))/unit
            HPL(3,j) = 4.1841*(-110238.9+(1.728*TL(j))+(4.8625*(TL(j)**2))
                        -(1.665E-05*(TL(j)**3))+(1.729E-09*(TL(j)**4))-lambda(3))/unit
            HPL(4,j) = 4.1841*(60135.845+(7.701*TL(j))+(2.2975-04*(TL(j)**2))
                        +(8.4E-07*(TL(j)**3))-(2.15E-10*(TL(j)**4))-lambda(4))/unit


      HV(j) = 0.
      HL(j) = 0.
      do i = 1, Comp
            HV(j) = HV(j) + y(i,j)*HPV(i,j)
```

```
            HL(j) = HL(j) + x(i,j)*HPL(i,j)
        end do

        ! CALCULATE ENTHALPY OF FEED

        if (TF(j) /= 0) then
            lambdaf(1) = 5600*(((1-(TF(j)/594.4))/0.3420)**0.38)
            lambdaf(2) = 9260*(((1-(TF(j)/516.2))/0.3191)**0.38)
            lambdaf(3) = 7700*(((1-(TF(j)/523.2))/0.3305)**0.38)
            lambdaf(4) = 9717*(((1-(TF(j)/647.3))/0.4235)**0.38)

            HPLf(1,j) = 4.1841*(-10607.89+(1.156*TF(j))+(3.0435E-02*(TF(j)**2))
                    -(1.3956E-05*(TF(j)**3))+(2.955E-09*(TF(j)**4))-
            lambdaf(1))/unit
            HPLf(2,j) = 4.1841*(-58855.738+(2.153*TF(j))+(2.5565-02*(TF(j)**2))
                    -(6.68E-06*(TF(j)**3))+(8.2E-11*(TF(j)**4))-lambdaf(2))/unit
            HPLf(3,j) = 4.1841*(-110238.9+(1.728*TF(j))+(4.8625*(TF(j)**2))
                    -(1.665E-05*(TF(j)**3))+(1.729E-09*(TF(j)**4))-lambdaf(3))/unit
            HPLf(4,j) = 4.1841*(60135.845+(7.701*TF(j))+(2.2975-04*(TF(j)**2))
                    +(8.4E-07*(TF(j)**3))-(2.15E-10*(TF(j)**4))-lambdaf(4))/unit

            VF(j) = 0.
            LF(j) = 0.
            do i = 1, Comp
                HLF(j) = HLF(j) + z(i,j)*HPLf(i,j)
            end do
        end if
    end do

end Subroutine Enthalpies
```

```
Subroutine Equation (n, Tvr, Tlr, Tir, SV, SL, fV1, fV2, fV3, fV4, fL1, fL2, fL3, fL4, U, N1, N2, N3, N4, &
                     r1, r2, r3, r4, QV, QL, l, v, xl, yl, x, y, z, TL, TV, Tl, TF, Vfr, Lfr, D, QC, QR, P, &
                     Fx, Comp, Limit, nEq, Tbw)

      implicit none
      integer, intent (in) :: n, nEq, Comp
      double precision, dimension (1:nEq,1) :: Fx
      double precision, dimension (1:n) :: HL, HV, HLF, HVF, Tvr, Tlr, Tir, haL, haV, MV1, MV2, &
                                           MV3, MV4, ML1, ML2, ML3, ML4, Vfr, Lfr, EV, EL, El, &
                                           RV1, RV2, RV3, RL1, RL2, RL3, Ql1, Ql2, Ql3, Ql4, SV, &
                                           SL, fV1, fV2, fV3, fV4, fL1,fL2, fL3, fL4, U, N1, N2, N3, &
                                           N4, r1, r2, r3, r4, QV, QL, M1, M2, M3, M4, E, Q1, Q2, &
                                           Q3, Q4, P, Sxl, Syl, TF, TL, TV, Tl, TT
      double precision, dimension (1:Comp,1:n) :: HPL, HPV, l, v, xl, yl, x, y, z, K, Kl, Psat, xx
      double precision, dimension (1:Comp,1:Comp) :: KkV, KkL
      double precision :: D, QC, QR, Tbw
      integer :: j, m, Comp, Limit

      call Enthalpies (n, HL, HV, HLF, HVF, x, y, z, TL, TV, P, TF, HPL, HPV, Comp)

      do j = 1, n
            call RxnRate (n, j, r1, r2, r3, r4, x, TL, Comp)

            if (j == 1) then

                  ! Total condenser
                        M1(j) = (1+(D/Lfr(j)))*l(1,j) - v(1,j+1) - r1(j)*U(j)
                        M2(j) = (1+(D/Lfr(j)))*l(2,j) - v(2,j+1) - r2(j)*U(j)
                        M3(j) = (1+(D/Lfr(j)))*l(3,j) - v(3,j+1) - r3(j)*U(j)
                        M4(j) = (1+(D/Lfr(j)))*l(4,j) - v(4,j+1) - r4(j)*U(j)
                        E(j) = (1+(D/Lfr(j)))*HL(j)*Lfr(j) + QC - HV(j+1)*Vfr(j+1)

            else if (j>1.and.j<n) then
                        call MTCoeff (TL, TV, Vfr, Lfr, P, xl, yl, x, y, j, n, Comp, KkV, KkL, Limit, &
                                           haV, haL)
```

```
     TT(j) = Tl(j)
     do m = 1, Comp
            xx(m,j) = xl(m,j)
     end do

     call PhaseEq (n, TT, P, Psat, xx, K, Kl, j, Comp, Limit)

! Column
     MV1(j) = (1+SV(j))*v(1,j) - v(1,j+1) - fV1(j) + N1(j)
     MV2(j) = (1+SV(j))*v(2,j) - v(2,j+1) - fV2(j) + N2(j)
     MV3(j) = (1+SV(j))*v(3,j) - v(3,j+1) - fV3(j) + N3(j)
     MV4(j) = (1+SV(j))*v(4,j) - v(4,j+1) - fV4(j) + N4(j)
     EV(j) = (1+SV(j))*HV(j)*Vfr(j) + QV(j) + haV(j)*Tbw*(Tvr(j)-Tir(j))
            - HV(j+1)*Vfr(j+1) - HVF(j)*(fV1(j)+fV2(j)+fV3(j)+fV4(j))
            + (N1(j)*HPV(1,j) + N2(j)*HPV(2,j) + N3(j)*HPV(3,j) +
            N4(j)*HPV(4,j))
     ML1(j) = (1+SL(j))*l(1,j) - l(1,j-1) - fL1(j) - N1(j) - r1(j)*U(j)
     ML2(j) = (1+SL(j))*l(2,j) - l(2,j-1) - fL2(j) - N2(j) - r2(j)*U(j)
     ML3(j) = (1+SL(j))*l(3,j) - l(3,j-1) - fL3(j) - N3(j) - r3(j)*U(j)
     ML4(j) = (1+SL(j))*l(4,j) - l(4,j-1) - fL4(j) - N4(j) - r4(j)*U(j)

     EL(j) = (1+SL(j))*HL(j)*Lfr(j)+QL(j) - haL(j)*Tbw*(Tir(j)-Tlr(j)) - HL(j-1)*Lfr(j-1)
            - HLF(j)*(fL1(j)+fL2(j)+fL3(j)+fL4(j)) - N1(j)*HPL(1,j)+N2(j)*HPL(2,j)
            + N3(j)*HPL(3,j) + N4(j)*HPL(4,j))
     EI(j) = haV(j)*Tbw*(Tvr(j)-Tir(j)) + (N1(j)*HPV(1,j) + N2(j)*HPV(2,j)
          + N3(j)*HPV(3,j) + N4(j)*HPV(4,j)) - haL(j)*Tbw*(Tir(j)-Tlr(j))
          - (N1(j)*HPL(1,j) + N2(j)*HPL(2,j) + N3(j)*HPL(3,j) + N4(j)*HPL(4,j))

     RV1(j) = N1(j) - (KkV(1,1)*(y(1,j)-yl(1,j))+KkV(1,2)*(y(2,j)-yl(2,j)))
            + KkV(1,3)*(y(3,j)-yl(3,j))-(y(1,j)*(N1(j)+N2(j)+N3(j)+N4(j))
     RV2(j) = N2(j) - (KkV(2,1)*(y(1,j)-yl(1,j)) + KkV(2,2)*(y(2,j)-yl(2,j))
            + KkV(2,3)*(y(3,j)-yl(3,j))-(y(2,j)*(N1(j)+N2(j)+N3(j)+N4(j))
     RV3(j) = N3(j) - (KkV(3,1)*(y(1,j)-yl(1,j)) + KkV(3,2)*(y(2,j)-yl(2,j))
            + KkV(3,3)*(y(3,j)-yl(3,j))-(y(3,j)*(N1(j)+N2(j)+N3(j)+N4(j))
     RL1(j) = N1(j) - (KkL(1,1)*(xl(1,j)-x(1,j)) + KkL(1,2)*(xl(2,j)-x(2,j))
            + KkL(1,3)*(xl(3,j)-x(3,j))-(x(1,j)*(N1(j)+N2(j)+N3(j)+N4(j))
```

```
          RL2(j) = N2(j) - (KkL(2,1)*(xl(1,j)-x(1,j)) + KkL(2,2)*(xl(2,j)-x(2,j))
                   + KkL(2,3)*(xl(3,j)-x(3,j))-(x(2,j)*(N1(j)+N2(j)+N3(j)+N4(j))
          RL3(j) = N3(j) - (KkL(3,1)*(xl(1,j)-x(1,j)) + KkL(3,2)*(xl(2,j)-x(2,j))
                   + KkL(3,3)*(xl(3,j)-x(3,j))-(x(3,j)*(N1(j)+N2(j)+N3(j)+N4(j))


          Ql1(j) = Kl(1,j)*xl(1,j) - yl(1,j)
          Ql2(j) = Kl(2,j)*xl(2,j) - yl(2,j)
          Ql3(j) = Kl(3,j)*xl(3,j) - yl(3,j)
          Ql4(j) = Kl(4,j)*xl(4,j) - yl(4,j)
          Sxl(j) = 1 - xl(1,j) - xl(2,j) - xl(3,j) - xl(4,j)
          Syl(j) = 1 - yl(1,j) - yl(2,j) - yl(3,j) - yl(4,j)


     else if (j == n) then


          TT(j) = TL(j)
          do m = 1, Comp
                   xx(m,j) = x(m,j)
          end do


          call PhaseEq (n, TT, P, Psat, xx, K, Kl, j, Comp, Limit)


       ! Reboiler
          Q1(j) = K(1,j)*x(1,j) - y(1,j)
          Q2(j) = K(2,j)*x(2,j) - y(2,j)
          Q3(j) = K(3,j)*x(3,j) - y(3,j)
          Q4(j) = K(4,j)*x(4,j) - y(4,j)
          M1(j) = v(1,j) + l(1,j) - l(1,j-1) - r1(j)*U(j)
          M2(j) = v(2,j) + l(2,j) - l(2,j-1) - r2(j)*U(j)
          M3(j) = v(3,j) + l(3,j) - l(3,j-1) - r3(j)*U(j)
          M4(j) = v(4,j) + l(4,j) - l(4,j-1) - r4(j)*U(j)
          E(j) = (HV(j)*Vfr(j) + HL(j)*Lfr(j) - HL(j-1)*Lfr(j-1) - QR)
     end if


     m = j
     if (m == 1) then
          Fx(1:5,1) = (/ M1(m), M2(m), M3(m), M4(m), E(m) /)
```

```
          else if (m > 1 .and. m < n)then
                    Fx(6+23*(m-2):5+23*(m-1),1) = (/ MV1(m), MV2(m), MV3(m), MV4(m), &
                                                      EV(m), EL(m), ML1(m), ML2(m), &
                                                      RV1(m), RV2(m), ML3(m), ML4(m), &
                                                      RL3(m), RV3(m), RL1(m), RL2(m), &
                                                      QI1(m), QI2(m), QI3(m), QI4(m), &
                                                      EI(m), SxI(m), SyI(m) /)
          else if (m == n) then
                    Fx(6+23*(m-2):5+23*(m-2)+9,1) = (/ M1(m), M2(m), M3(m), M4(m), &
                                                        Q1(m), Q2(m), Q3(m), Q4(m), E(m) /)
          end if

     end do

end Subroutine Equation



Subroutine Feed (n, TF, z, P, fL1, fL2, fL3, fL4, Comp, Limit)

     implicit none
     integer, intent (in) :: n, Comp
     double precision, dimension (1:n) :: fV1, fV2, fV3, fV4, fL1, fL2, fL3, fL4, TF, feedL, feedV, &
                                          TT, P
     integer, dimension (1:n) :: fstage
     double precision, dimension(1:Comp,1:n) :: xx, Psat, K, z
     double precision :: Tfeed, Tav
     integer :: i, j, q, Limit, nfeed

     do j = 1, n
          fL1(j) = 0.
          fL2(j) = 0.
          fL3(j) = 0.
          fL4(j) = 0.
     end do
     nfeed = 1
```

```
    do q = 1, nfeed
            fstage(q) = 3
            ! sat'd liquid feed
            feedL(fstage(q)) = 0.2584/60                    ! mol/s
            feedV(fstage(q)) = 0.0000
             z(1,fstage(q))     = 0.2559                                ! acetic acid
             z(2,fstage(q))     = 0.6159                                ! ethanol
             z(3,fstage(q))     = 0.0539                                ! ethyl acetate
             z(4,fstage(q))     = 0.0743                                ! water
            fL1(fstage(q)) = z(1,fstage(q))*feedL(fstage(q))
            fL2(fstage(q)) = z(2,fstage(q))*feedL(fstage(q))
            fL3(fstage(q)) = z(3,fstage(q))*feedL(fstage(q))
            fL4(fstage(q)) = z(4,fstage(q))*feedL(fstage(q))
            fV1(fstage(q)) = z(1,fstage(q))*feedV(fstage(q))
            fV2(fstage(q)) = z(2,fstage(q))*feedV(fstage(q))
            fV3(fstage(q)) = z(3,fstage(q))*feedV(fstage(q))
            fV4(fstage(q)) = z(4,fstage(q))*feedV(fstage(q))

            do j = 1, n
                    TF(j) = 0.
                    if ( j == fstage(q)) then
                            do i = 1, Comp
                                    xx(i,j) = z(i,j)
                            end do
                            call Bubble_Point (n, P, xx, Psat, K, TT, Tav, j, Comp, Limit)
                            TF(fstage(q)) = Tav
                    end if
            end do
    end do
    Tfeed = 0.
    do q = 1, nfeed
            Tfeed = Tfeed + feedL(fstage(q))
    end do

end Subroutine Feed
```

```
Subroutine HTCoeff (n, j, haL, haV, kaVave, kaLave, CpVm, CpLm, TcVm, TcLm, DVave, DLave, &
                    denLm, denV)

     implicit none
     integer, intent (in) :: n
     double precision, dimension (1:n) :: haL, haV, CpVm, CpLm, TcVm, TcLm, DVave, DLave, &
                                          kaVave, kaLave,  denLm, denV
     double precision, parameter :: unit = 1000000
     integer :: j

     ! Heat transfer coefficient, MJ/s K

     haV(j) = (kaVave(j)*CpVm(j)*((TcVm(j)/(denV(j)*CpVm(j)*DVave(j)))**(2/3)))/unit
     haL(j) = (kaLave(j)*CpLm(j)*((TcLm(j)/(denLm(j)*CpLm(j)*DLave(j)))**0.5))/unit

end Subroutine HTCoeff


Subroutine MTCoeff (TL, TV, Vfr, Lfr, P, xl, yl, x, y, j, n, Comp, KkV, KkL, Limit, haV, haL)

     implicit none
     integer, intent (in) :: n, Comp
     double precision, dimension(1:Comp) :: visL, visV, ac, ac1, denL
     double precision, dimension(1:Comp,1:Comp) :: DL, DV, RL, RV, RP, KkP, dirac, gamma, &
                                                   ac2, KkV, KkL, kaL, kaV
     double precision, dimension (1:n) :: TV, TL, P, denV, Vfr, Lfr, denLm, TT, kaVave, kaLave, &
                                          DVave, DLave, CpVm, CpLm, TcVm, TcLm, haV, haL, &
                                          denVm, dummyL, dummyV, dummyTL, dummyTV
     double precision, dimension(1:Comp,1:n) :: xl, yl, x, y, xav, yav, xx, CpV, CpL, TcV, TcL
     double precision, parameter :: Wx = 0.015, ZL = 0.07, Ab = 0.014
     double precision :: Fs, Lf, tal, Zc, us, h
     integer :: i, j, m, mm, qq, Comp, Limit
     intrinsic Exp, Log, Matmul

     call Phys_Chem_Prop(denL, denLm, denV, visL, visV, CpV, CpL, TcV, TcL, TL, TV, P, x, y, &
                         j, n, CpVm, CpLm, TcVm, TcLm, Comp)
```

```
if (j > 1 .and. j < n) then
    call Diffusivity (DL, DV, TL, TV, P, visL, j, n, DLave, DVave)

  ! Bubble Cap Tray

    us = Vfr(j)*(60.05*y(1,j)+46.068*y(2,j)+88.10*y(3,j)+18.016*y(4,j))/1000        ! kg/s
    Fs = us*((denV(j)*(60.05*y(1,j)+46.068*y(2,j)+88.10*y(3,j)+18.016*y(4,j)))**0.5)/(Ab)
                                                                        ! kg^0.5/m^0.5 s
    Lf = Lfr(j)/(0.11*denLm(j)*1000)                ! m2/s
    Zc = 0.0254*(1.65+7.4803*Wx+80.478*Lf-0.5328*Fs)
    tal = 1.2002*Zc*ZL/Lf
    denVm(j) = denV(j)*(60.05*y(1,j)+46.068*y(2,j)+88.10*y(3,j)+18.016*y(4,j))

    do i = 1, Comp
        do m = 1, Comp
            kaV(i,m) = (0.776+4.567*Wx-0.2377*Fs+87.319*Lf)*Vfr(j)
                       /((visV(i)/(denVm(j)*DV(i,m)))**0.5)
            kaL(i,m) = 20308*(DL(i,m)**0.5)*(0.21313*Fs+0.15)*tal*Lfr(j)
        end do
    end do
    kaVave(j) = 0.
    kaVave(j) = (kaV(1,2)+kaV(1,3)+kaV(1,4)+kaV(2,3)+kaV(2,4)+kaV(3,4))/6
    kaLave(j) = 0.
    kaLave(j) = (kaL(1,2)+kaL(1,3)+kaL(1,4)+kaL(2,3)+kaL(2,4)+kaL(3,4))/6
    do i = 1, Comp
        yav(i,j) = (y(i,j)+yI(i,j))/2
        xav(i,j) = (x(i,j)+xI(i,j))/2
    end do

    ! Mass Transfer Coefficient in Vapor Phase, kmol/s
    do i = 1, Comp-1
        do m = 1, Comp-1
            if (i /= m) then
                RV(i,m) = -yav(i,j)*((1/kaV(i,m))-(1/kaV(i,Comp)))
            else
                RV(i,m) = yav(i,j)/kaV(i,Comp)
```

```
            do qq = 1, Comp
                if (qq /= i) then
                    RV(i,m) = RV(i,m +(yav(qq,j)/kaV(i,qq))
                else
                    RV(i,m) = RV(i,m)
                end if
            end do
        end if
    end do
end do

do i = 1, Comp-1
    do m = 1, Comp-1
        RP(i,m) = RV(i,m)
    end do
end do
call InvsMatrix (RP, KkP, Comp)
do i = 1, Comp-1
    do m = 1, Comp-1
        KkV(m,i) = KkP(m,i)
    end do
end do

!Mass Transfer Coefficient in Liquid Phase
do i = 1, Comp-1
    do m = 1, Comp-1
        if (i /= m) then
            RL(i,m) = -xav(i,j)*((1/kaL(i,m))-(1/kaL(i,Comp)))
        else
            RL(i,m) = xav(i,j)/kaL(i,Comp)
            do qq = 1, Comp
                if (qq /= i) then
                    RL(i,m) = RL(i,m) + (xav(qq,j)/kaL(i,qq))
                else
                    RL(i,m) = RL(i,m)
                end if
```

```
                        end do
                end if
        end do
end do

do i = 1, Comp-1
        do m = 1, Comp-1
                RP(i,m) = RL(i,m)
        end do
end do
call InvsMatrix (RP, KkP, Comp)
do i = 1, Comp
        xx(i,j) = 0.0
        xx(i,j) = xav(i,j)
        TT(j) = TL(j)
end do
call UNIFAC (xx, n, TT, ac, j, Comp, Limit)
do i = 1, Comp
        ac1(i) = ac(i)
end do
do mm = 1, Comp
        do i = 1, Comp
                if (i == mm) then
                        xx(i,j) = 1.0001*xx(i,j)
                else
                        xx(i,j) = xx(i,j)
                end if
        end do
        call UNIFAC (xx, n, TT, ac, j, Comp, Limit)
        do i = 1, Comp
                ac2(i,mm) = ac(i)
                if (i == mm) then
                        xx(i,j) = xx(i,j)/1.0001
                else
                        xx(i,j) = xx(i,j)
```

```
                    end if
                end do
            end do

            do i = 1, Comp-1
                do mm = 1, Comp-1
                    if (i == mm) then
                        dirac(i,mm) = 1
                    else
                        dirac(i,mm) = 0
                    end if
                    gamma(i,mm) = dirac(i,mm) + xx(i,j)*(log(ac1(i))-log(ac2(i,mm)))
                                   /(1.0001*xx(mm,j)-xx(mm,j))
                end do
            end do

            KkL = Matmul(KkP,gamma)
            call HTCoeff (n, j, haL, haV, kaVave, kaLave, CpVm, CpLm, TcVm, TcLm, DVave, &
                          DLave, denLm, denV)
        end if

end Subroutine MTCoeff


Subroutine ObjectiveF (n, nEq, jp, Ind_var, SV, SL, fV1, fV2, fV3, fV4, fL1, fL2, fL3, fL4, U, QV, QL, &
                       Vfr, Lfr, P, v, l, x, y, z, Func, Tbw, TF, D, QC, QR, Comp, Limit, np, Fx)

        implicit none
        integer, intent (in) :: n, nEq, np
        double precision, dimension (1:nEq,0:np) :: Ind_var
        double precision, dimension (0:np) :: Func
        double precision, dimension (1:nEq,1) :: Fx
        double precision :: Tbw, D, QC, QR, sumxl, sumyl
        double precision, dimension (1,1) :: SSE
        double precision, dimension (1:n) :: Tvr, Tlr, Tir, TL, TV, Tl, SV, SL, fV1, fV2, fV3, fV4, fL1, &
```

```
                                        fL2, fL3, fL4, U, N1, N2, N3, N4, r1, r2, r3, r4, QV, QL, &
                                        Vfr, Lfr, P, TF, Vtot, Ltot
double precision, dimension (1:Comp,1:n) :: v, l, x, y, z, xl, yl
integer :: i, m, Comp, Limit, jp, j

do i = 1, Comp
        l(i,1) = Ind_var(i,jp)
end do
Tlr(1) = Ind_var(5,jp)

do m = 2, n-1
        do i = 1, Comp
                v(i,m) = Ind_var(5+i+23*(m-2),jp)
        end do
        Tvr(m) = Ind_var(10+23*(m-2),jp)
        Tlr(m) = Ind_var(11+23*(m-2),jp)
        do i = 1, Comp
                l(i,m) = Ind_var(11+i+23*(m-2),jp)
        end do
        N1(m) = Ind_var(16+23*(m-2),jp)
        N2(m) = Ind_var(17+23*(m-2),jp)
        N3(m) = Ind_var(18+23*(m-2),jp)
        N4(m) = Ind_var(19+23*(m-2),jp)
        xl(3,m) = Ind_var(20+23*(m-2),jp)
        xl(2,m) = Ind_var(21+23*(m-2),jp)
        xl(1,m) = Ind_var(22+23*(m-2),jp)
        yl(2,m) = Ind_var(23+23*(m-2),jp)
        yl(3,m) = Ind_var(24+23*(m-2),jp)
        yl(1,m) = Ind_var(25+23*(m-2),jp)
        Tir(m) = Ind_var(26+23*(m-2),jp)
        xl(4,m) = Ind_var(27+23*(m-2),jp)
        yl(4,m) = Ind_var(28+23*(m-2),jp)
end do

m = n
```

```
do i = 1, Comp
        v(i,m) = Ind_var(5+23*(m-2)+i,jp)
end do
Tlr(m) = Ind_var(5+23*(m-2)+5,jp)
Tvr(m) = Tlr(m)
do i = 1, Comp
        l(i,m) = Ind_var(5+23*(m-2)+5+i,jp)
end do
do j = 1, n
        TL(j) = Tbw*Tlr(j)
        TV(j) = Tbw*Tvr(j)
        Tl(j) = Tbw*Tir(j)
end do

TL(n) = TV(n)

do j = 2, n-1
        sumxl = xl(1,j)+xl(2,j)+xl(3,j)+xl(4,j)
        sumyl = yl(1,j)+yl(2,j)+yl(3,j)+yl(4,j)

        do i = 1, Comp
                xl(i,j) = xl(i,j)/sumxl
                yl(i,j) = yl(i,j)/sumyl
        end do
end do
do j = 1, n
        Vtot(j) = 0.
        Ltot(j) = 0.
        do i = 1, Comp
                Vtot(j) = Vtot(j) + v(i,j)
                Ltot(j) = Ltot(j) + l(i,j)
        end do
        do i = 1, Comp
                if (Vtot(j) /= 0.) then
                        v(i,j) = v(i,j)*Vfr(j)/Vtot(j)
```

```
                        y(i,j) = v(i,j)/Vfr(j)
                  end if
                  if (Ltot(j) /= 0.) then
                        l(i,j) = l(i,j)*Lfr(j)/Ltot(j)
                        x(i,j) = l(i,j)/Lfr(j)
                  end if
            end do
      end do

      call Equation (n, Tvr, Tlr, Tir, SV, SL, fV1, fV2, fV3, fV4, fL1, fL2, fL3, fL4, U, N1, N2, N3, &
                  N4, r1, r2, r3, r4, QV, QL, l, v, xl, yl, x, y, z, TL, TV, Tl, TF, Vfr, Lfr, D, QC, &
                  QR, P, Fx, Comp, Limit, nEq, Tbw)

      SSE = (Matmul(Transpose(Fx),Fx))
      Func(jp) = SSE(1,1)

end Subroutine ObjectiveF



Subroutine PhaseEq (n, TT, P, Psat, xx, K, Kl, j, Comp, Limit)

      implicit none
      integer, intent (in) :: n, Comp
      double precision, dimension (1:n) :: P, TT
      double precision, dimension(1:Comp,1:n) :: xx, K, Kl, Psat
      double precision, dimension(1:Comp) :: A, B, C, ac
      integer :: i, j, Limit

      ! Calculate Psat(atm) from Antoine equation
      A(1:4) = (/7.18807, 8.04494, 7.09808, 7.96681/)
      B(1:4) = (/1416.7, 1554.3, 1238.71, 1668.21/)
      C(1:4) = (/211.0, 222.65, 217.0, 228.0/)

      if (j == 1 .or. j == n) then
            do i = 1, Comp
                  Psat(i,j) = (10**(A(i) - (B(i)/((TT(j)-273.13) + C(i)))))/760
```

```
            end do
        else
            do i = 1, Comp
                Psat(i,j) = (10**(A(i) - (B(i)/((TT(j)-273.13) + C(i)))))/760
            end do
        end if
        call UNIFAC (xx, n, TT, ac, j, Comp, Limit)
        do i = 1, Comp
            KI(i,j) = ac(i)*Psat(i,j)/P(j)
        end do
        if (j == 1 .or. j == n) then
            do i = 1, Comp
                K(i,j) = KI(i,j)
            end do
        end if

    end Subroutine PhaseEq


    Subroutine Phys_Chem_Prop (denL, denLm, denV, visL, visV, CpV, CpL, TcV, TcL, TL, TV, P, x, y, &
                                    j, n, CpVm, CpLm, TcVm, TcLm, Comp)

        implicit none
        integer, intent (in) :: n, Comp
        double precision, dimension (1:Comp) :: denL, visL, visV, Ts, sigma, omv, del, Ms
        double precision, dimension (1:Comp,1:n) :: x, y, CpV, CpL, TcV, TcL
        double precision, dimension (1:n) :: TV, TL, P, denV, denLm, CpVm, CpLm, TcVm, TcLm
        double precision, parameter :: unit = 1000000
        integer :: i, j, Comp

        ! Molar Density (kmol/m^3)

          ! liquid
            denL(1) = 17.4688
            denL(2) = 17.1269
            denL(3) = 10.2270
```

```
    denL(4) = 55.5062

! liquid mixture
    denLm(j) = (x(1,j)*denL(1)+x(2,j)*denL(2)+x(3,j)*denL(3)+x(4,j)*denL(4))

    if (j > 1) then

        ! vapor (kmol/m3)

            denV(j) = 12.1862*P(j)/TV(j)                    ! P = atm, TV = K

        ! Liquid Viscosity , (cP)

            visL(1) = (4E-10)*(TL(j)**4) - (8E-7)*(TL(j)**3) + 0.0006*(TL(j)**2)
                      - 0.1913*TL(j) + 24.755
            visL(2) = (8E-9)*(TL(j)**4) -(1E-5)*(TL(j)**3) + 0.0061*(TL(j)**2)
                      - 1.4624*TL(j) + 134.93
            visL(3) = -(7E-8)*(TL(j)**3) + (1E-4)*(TL(j)**2) - 0.0432*TL(j) + 6.6604
            visL(4) = 0.0002*(TL(j)**2) - 0.1253*TL(j) + 23.091


    ! Vapor Viscosity , (cP)
            VisV(1) = (-(1E-7)*(TV(j)**3) + 0.0004*(TV(j)**2) + 0.1756*TV(j)
                        - 22.491)/unit
            VisV(2) = ((1E-7)*(TV(j)**3) - 0.0002*(TV(j)**2) + 0.3945*TV(j) - 11.688)/unit
            VisV(3) = (-(3E-20)*(TV(j)**3) - (5E-5)*(TV(j)**2) + 0.2822*TV(j)
                        - 2.9249)/unit
            VisV(4) = (-(4E-6)*(TV(j)**2) + 0.4088*TV(j) - 29.024)/unit

    ! Heat Capacity of Gas, kJ/kmol K
            CpV(1,j) = ((-1E-06)*(TV(j)**2) + 0.0034*TV(j) + 0.1582)*60.053
            CpV(2,j) = ((-1E-06)*(TV(j)**2) + 0.0043*TV(j) + 0.2445)*46.069
            CpV(3,j) = ((-1E-06)*(TV(j)**2) + 0.0037*TV(j) + 0.3201)*88.106
            CpV(4,j) = ((2E-07)*(TV(j)**2) + 0.0004*TV(j) + 1.7361)*18.015
            CpVm(j) = 0.
            do i = 1, Comp
                    CpVm(j) = CpVm(j) + y(i,j)*CpV(i,j)
            end do
```

```
        ! Heat Capacity of Liquid, kJ/kmol K
            CpL(1,j) = (0.0041*TL(j) + 0.8166)*60.053
            CpL(2,j) = ((4E-05)*(TL(j)**2) - 0.0163*TL(j) + 3.4196)*46.069
            CpL(3,j) = ((1E-05)*(TL(j)**2) - 0.0051*TL(j) + 2.1425)*88.106
            CpL(4,j) = ((5E-10)*(TL(j)**4) - (8E-07)*(TL(j)**3) + 0.0005*(TL(j)**2)
                 - 0.1171*TL(j) + 15.542)*18.015
            CpLm(j) = 0.
            do i = 1, Comp
                CpLm(j) = CpLm(j) + x(i,j)*CpL(i,j)
            end do

        ! Thermal Conductivity of Gas, kW/m K
            TcV(1,j) = ((1E-07)*(TV(j)**2) - (8E-07)*TV(j) + 0.0017)/1000
            TcV(2,j) = ((7E-08)*(TV(j)**2) + (7E-05)*TV(j) - 0.0114)/1000
            TcV(3,j) = ((1E-07)*(TV(j)**2) - (5E-06)*TV(j) + 0.0028)/1000
            TcV(4,j) = ((4E-08)*(TV(j)**2) + (5E-05)*TV(j) - 0.0007)/1000

            TcVm(j) = 0.
            do i = 1, Comp
                TcVm(j) = TcVm(j) + y(i,j)*TcV(i,j)
            end do

        ! Thermal Conductivity of Liquid, kW/m K
            TcL(1,j) = (-0.0002*TL(j) + 0.2139)/1000
            TcL(2,j) = (-0.0003*TL(j) + 0.2524)/1000
            TcL(3,j) = (-0.0004*TL(j) + 0.2493)/1000
            TcL(4,j) = ((-7E-09)*(TL(j)**3) - (3E-06)*(TL(j)**2) + 0.005*TL(j)
                 - 0.4726)/1000
            TcLm(j) = 0.
            do i = 1, Comp
                TcLm(j) = TcLm(j) + x(i,j)*TcL(i,j)
            end do
        end if

end Subroutine Phys_Chem_Prop
```

```
Subroutine Residual (ll, mvp, marp, sf, af, xx, Tg, sec, lp, ng, lacr, lacc, TT, i, j, m, n, qq, Comp, limit)


        implicit none
        integer, intent(in) :: n
        double precision, intent(inout), dimension(1:n) :: TT
        double precision, intent(inout), dimension(1:4,1:n) :: xx
        double precision, dimension (1:47,1:47) :: lp,Tg
        integer, dimension (1:90,1:90) :: ng, sec
        integer, dimension(1:90) :: MG
        double precision, dimension (1:90,1:90) :: xfc, afc, frac5, frac6, lgrp
        double precision, dimension(1:90) :: ll, mvp, marp, sf, af, lacc, xfm, afm, var1, var2, &
                                             dummy1, dummy2, R, Q, MW, frac3, frac4, lgr, lacr
        double precision :: frac, frac1, frac2
        integer :: i, j, Comp, m, qq, limit
        intrinsic Exp, Log

        do i = 1, Comp
                dummy1(i) = 0.
                do qq = 1, limit
                        dummy1(i) = dummy1(i) + ng(i,qq)
                end do
                do qq = 1, limit
                        xfc(i,qq) = ng(i,qq)/dummy1(i)
                end do
        end do
        call Group_Spec (MG,R,Q,MW)
        do i = 1, Comp
                dummy2(i) = 0.
                do qq = 1, limit
                        dummy2(i) = dummy2(i) + xfc(i,qq)*Q(qq)
                end do
                do qq = 1, limit
                        afc(i,qq) = xfc(i,qq)*Q(qq)/dummy2(i)
                end do
        end do
```

```
do m = 1, Comp
          do qq = 1, limit
                    frac5(m,qq) = 0
                    do i = 1, limit
                              if (afc(m,i) /= 0.) then
                                        call Interact_parameter (Ip)
                                        Tg(MG(i),(MG(qq))) = Exp(-Ip(MG(i),(MG(qq)))/TT(j))
                                        frac5(m,qq) = frac5(m,qq) + afc(m,i)*Tg(MG(i),(MG(qq)))
                              end if
                    end do
          end do
end do
do m = 1, Comp
          do qq = 1, limit
                    frac6(m,qq) = 0.
                    do i = 1, limit
                              if (afc(m,i) /= 0.) then
                                        call Interact_parameter (Ip)
                                        Tg(MG(qq),(MG(i))) = Exp(-Ip(MG(qq),(MG(i)))/TT(j))
                                        frac6(m,qq) = frac6(m,qq) + (afc(m,i)*Tg(MG(qq),MG(i))
                                                                     /frac5(m,i))
                              end if
                    end do
          end do
end do
do m = 1, Comp
          do qq = 1, limit
          lgrp(m,qq) = 0.
                    if (afc(m,qq) /= 0.) then
                    call Group_Spec (MG,R,Q,MW)
                    lgrp(m,qq) = Q(qq)*(1-(log(frac5(m,qq)))-frac6(m,qq))
                    end if
          end do
end do
```

```
frac1 = 0.
do i = 1, Comp
        var2(i) = 0.
        do qq = 1, limit
                var2(i) = var2(i) + ng(i,qq)
        end do
        frac1 = frac1 + xx(i,j)*var2(i)
end do
do qq = 1, limit
        var1(qq) = 0.
        do i = 1, Comp
                var1(qq) = var1(qq) + ng(i,qq)*xx(i,j)
        end do
end do
do qq = 1, limit
        xfm(qq) = var1(qq)/frac1
end do
call Group_Spec (MG,R,Q,MW)
frac2 = 0.
do qq = 1, limit
        frac2 = frac2 + xfm(qq)*Q(qq)
end do

do qq = 1, limit
        do i = 1, Comp
                afm(sec(i,qq)) = xfm(sec(i,qq))*Q(sec(i,qq))/frac2
        end do
end do
do qq = 1, limit
        frac3(qq) = 0.
        do i = 1, limit
                if (afm(i) /= 0.) then
                        call Interact_parameter (Ip)
                        Tg(MG(i),(MG(qq))) = Exp(-Ip(MG(i),(MG(qq)))/TT(j))
                        frac3(qq) = frac3(qq) + afm(i)*Tg(MG(i),(MG(qq)))
```

```
                end if
            end do
        end do
        do qq = 1, limit
            frac4(qq) = 0.
            do i = 1, limit
                if (afm(i) /= 0.) then
                    call Interact_parameter (Ip)
                    Tg(MG(qq),(MG(i))) = Exp(-Ip(MG(qq),(MG(i)))/TT(j))
                    frac4(qq) = frac4(qq) + (afm(i)*Tg(MG(qq),MG(i))/frac3(i))
                end if
            end do
        end do
        do qq = 1, limit
            lgr(qq) = 0.
            if (afm(qq) /= 0.) then
                call Group_Spec (MG,R,Q,MW)
                lgr(qq) = Q(qq)*(1-(log(frac3(qq)))-frac4(qq))
            end if
        end do
        do i = 1, Comp
            lacr(i) = 0
            do qq = 1, limit
                if (ng(i,qq) /= 0) then
                    lacr(i) = lacr(i) + ng(i,qq)*(lgr(qq) - lgrp(i,qq))
                end if
            end do
        end do

        call Combinatorial (xx, ll, mvp, marp, sf, af, lacc, frac1, frac2, frac, i, j, Comp, n)

    end Subroutine Residual
```

```
Subroutine RxnRate (n, j, r1, r2, r3, r4, x, Lfr, Comp)

        implicit none
        integer, intent (in) :: n, j, Comp
        double precision, dimension (1:n) :: sumlV, r1, r2, r3, r4, C1, C2, C3, C4, Lfr
        double precision, dimension (1:Comp,1:n) :: x
        double precision, dimension (1:Comp) :: denL
        double precision :: rate

        ! Liquid Molar Density, mol/l

        denL(1) = 17.4688
        denL(2) = 17.1269
        denL(3) = 10.2270
        denL(4) = 55.5062

        C1(j) = x(1,j)*denL(1)
        C2(j) = x(2,j)*denL(2)
        C3(j) = x(3,j)*denL(3)
        C4(j) = x(4,j)*denL(4)

        ! Rate of Reaction, mol/l s

        rate = 0.
        rate =  ((4.76E-4)*C1(j)*C2(j) - (1.63E-4)*C3(j)*C4(j))/60

        r3(j) = rate
        r4(j) = r3(j)
        r1(j) = - r3(j)
        r2(j) = r1(j)

end Subroutine RxnRate
```

```
Subroutine UNIFAC (xx, n, TT, ac, j, Comp, Limit)

        implicit none
        integer, intent (in) :: n, Comp
        double precision, dimension (1:n) :: TT
        double precision, dimension (1:Comp,1:n) :: xx
        double precision, dimension (1:47,1:47) :: Ip, Tg
        double precision, dimension (1:90,1:90) :: xfc, afc, frac5, frac6, lgrp
        double precision, dimension(1:90) :: R, Q, MW, xfm, afm, mvp, marp, dummy1, dummy2, &
                                    lgr, var1, var2, sf, af, lac, lacc, lacr, ll, frac3, frac4
        double precision, dimension(1:Comp) :: ac
        double precision :: frac, frac1, frac2
        character(len=15), dimension(1:Comp,1:3) :: Func
        integer, dimension(1:90) :: MG, nf
        integer, dimension (1:90,1:90) :: ng, sec
        integer :: i, j, Comp, m, qq, Limit

        ! Number of functional group in each component.
        nf(1) = 2                                       ! Acetic Acid
        nf(2) = 3                                       ! Ethanol
        nf(3) = 3                                       ! Ethyl Acetate
        nf(4) = 1                                       ! Water

        Func(1,1) = 'CH3'           ! Func(order of comp.,order of func.gr.)
        Func(1,2) = 'COOH'
        Func(2,1) = 'CH3'
        Func(2,2) = 'CH2'
        Func(2,3) = 'OH'
        Func(3,1) = 'CH3'
        Func(3,2) = 'CH2'
        Func(3,3) = 'COO'
        Func(4,1) = 'H2O'

        sec(1,1) = 1
        sec(1,2) = 43
        sec(2,1) = 1
```

```
    sec(2,2) = 2
    sec(2,3) = 15
    sec(3,1) = 1
    sec(3,2) = 2
    sec(3,3) = 77
    sec(4,1) = 17

    ng(1,sec(1,1)) = 1
    ng(1,sec(1,2)) = 1
    ng(2,sec(2,1)) = 1
    ng(2,sec(2,2)) = 1
    ng(2,sec(2,3)) = 1
    ng(3,sec(3,1)) = 2
    ng(3,sec(3,2)) = 1
    ng(3,sec(3,3)) = 1
    ng(4,sec(4,1)) = 1

    call Group_Spec (MG, R, Q, MW)
    do i = 1, Comp
        mvp(i) = 0.
        marp(i) = 0.
        do qq = 1, limit
            mvp(i) = mvp(i) + ng(i,qq)*R(qq)
            marp(i) = marp(i) + ng(i,qq)*Q(qq)
        end do
    end do
    call Residual (ll, mvp, marp, sf, af, xx, Tg, sec, lp, ng, lacr, lacc, TT, i, j, m, n, qq, Comp, &
                limit)
    do i = 1, Comp
        ac(i) = 0
        lac(i) = lacc(i) + lacr(i)
        ac(i) = Exp(lac(i))
    end do

end Subroutine UNIFAC
```

AUTHOR BIOGRAPHY

Miss Walairat Suksukmai was born on March 21, 1976 in Nakornpathom. She graduated with a Bachelor Degree in Chemical Engineering from Thammasat University in 1997.

In 1998, she worked as an engineer in Bangkok Rubber Company Limited at Ayuttaya.