

CONTROL AND SIMULATION STUDIES FOR A MULTICOMPONENT BATCH PACKED  
DISTILLATION COLUMN

A THESIS SUBMITTED TO  
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES  
OF  
MIDDLE EAST TECHNICAL UNIVERSITY

BY

HATİCE CEYLAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR  
THE DEGREE OF MASTER OF SCIENCE  
IN  
CHEMICAL ENGINEERING

AUGUST 2007

Approval of the thesis:

**CONTROL AND SIMULATION STUDIES FOR A MULTICOMPONENT BATCH  
PACKED DISTILLATION COLUMN**

Submitted by **HATİCE CEYLAN** in partial fulfillment of the requirements for the degree of  
**Master of Science in Chemical Engineering Department, Middle East Technical  
University** by,

Prof. Dr. Canan Özgen  
Dean, Graduate School of **Natural and Applied Sciences**

---

Prof. Dr. Nurcan Baç  
Head of Department, **Chemical Engineering**

---

Prof. Dr. Canan Özgen  
Supervisor, **Chemical Engineering Department, METU**

---

**Examining Committee Members:**

Prof. Dr. Nazife Suzan Kincal  
Chemical Engineering Department, METU

---

Prof. Dr. Canan Özgen  
Chemical Engineering Department, METU

---

Prof. Dr. Kemal Leblebicioğlu  
Electrical and Electronics Engineering Department, METU

---

Prof. Dr. Gürkan Karakaş  
Chemical Engineering Department, METU

---

Assist. Prof. Dr. Yusuf Uludağ  
Chemical Engineering Department, METU

---

**Date:** 29.08.2007

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Hatice CEYLAN

Signature :

## **ABSTRACT**

### CONTROL AND SIMULATION STUDIES FOR A MULTICOMPONENT BATCH PACKED DISTILLATION COLUMN

Ceylan, Hatice

M. S., Department of Chemical Engineering

Supervisor: Prof. Dr. Canan Özgen

August 2007, 150 pages

During the last decades, batch distillation is preferably used with an increasing demand over continuous one, to separate fine chemicals in chemical and petroleum industries, due to its advantages like, flexibility and high product purity. Consequently, packed distillation columns, with newly generated packing materials, are advantageous compared to plate columns because of their smaller holdups, resistivity to corrosive materials and their higher separation efficiencies. Also, in many industrial applications, mathematical models of distillation systems are frequently used in order to design effective control systems, to train operating personnel and to handle fault diagnostics. Thus, the main objective of this study is to develop a mathematical model for a multicomponent batch distillation column, which is used to separate mixtures at low operating pressures, packed with random packing materials. In multicomponent batch packed distillation, operation with optimum reflux ratio profile is important for efficiency to maximize the amount of the distillate with a specified concentration, for a given time. Therefore, it is also aimed to find the optimum reflux ratio profile for the multicomponent batch packed distillation column.

A simulation algorithm is written with the aid of MATLAB and FORTRAN programming languages by taking into account pressure drop and variation of physical properties. The

selected incremental bed height,  $\Delta z$ , to be used in the simulation program has an effect on the accuracy of the results. This is analyzed and the optimal incremental height is found to be 3.5 cm for a 1.5m bed height. The change in distillate compositions with a given constant reflux ratio is found to be similar with those of previous studies. The simulation code is also used to obtain responses in distillate compositions for different reflux ratios, condenser holdups and reboiler duties and compared with similar studies found from literature and found to be adequate. Finally, experiments are conducted to verify simulation algorithm by using a lab-scale packed distillation column for the separation of a polar mixture of ethanol and water. It is observed that, there is a good agreement between the experimental and simulation results.

After the verification of dynamic model, optimum operation policy to maximize product amount is investigated numerically by using capacity factor approach. The column is operated with and without recycling of the holdups of the slop cut tanks, in order to examine the effect of recycling on capacity factor, CAP. It is observed that, recycling of the molar holdups of the slop cut tanks is resulted in a 28% increase in the separation efficiency.

Keywords: Multicomponent Batch Distillation System, Dynamic Modeling of Packed Columns, Optimal Operation

# ÖZ

## ÇOK BİLEŞENLİ KESİKLİ BİR DOLGULU DAMITMA KOLONUNDA KONTROL VE BENZETİM ÇALIŞMALARI

Ceylan, Hatice

Yüksek Lisans, Kimya Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Canan Özgen

Ağustos 2007, 150 sayfa

Son yıllarda, kimya ve petrol endüstrilerinde, özel kimyasalları ayırmak için sürekli damıtmaya oranla, işletmede gösterdiği esneklik ve elde edilen yüksek ürün saflığı nedenleri ile kesikli damıtma tercih edilmektedir. Özellikle, endüstride, dolgulu damıtma kolonları yeni geliştirilmiş dolgu maddeleri ile az sıvı tutma, korozif maddelere karşı dayanıklılık ve yüksek ayırtırmadaki etkin olma özellikleri ile tablalı kolonlardan daha fazla kullanılmaktadırlar. Ayrıca, birçok endüstriyel uygulamada, etkili kontrol sistemleri tasarlamak, personel eğitimi ve arıza tanımlamaları için matematiksel damıtma modelleri yoğun olarak kullanılmaktadır. Bu nedenle, bu çalışmanın en önemli amacı, düşük basınçlarda işletilen karışımları ayırmak için, rasgele yerleştirilmiş dolgu maddeli çok bileşenli kesikli-dolgulu damıtma kolonu için matematiksel bir model geliştirmektir. Ayrıca, bu çok bileşenli kesikli damıtma kolonunda, belirli bir zamanda, damıtılan ürün miktarını maksimize etmek üzere optimum geri akış profilinin belirlenmesi amaçlanmıştır.

Kolon için MATLAB ve FORTRAN programlama dillerini kullanarak, basınç düşmesi ve fiziksel özelliklerin değişimi dikkate alınarak, kolon değişkenlerinin dinamik değişimini belirlemek üzere, bir benzetim algoritması hazırlanmıştır. Dolgulu kolonda benzetim algoritmasının çalıştırılmasında seçilen dolgulu yatak adım yüksekliği,  $\Delta z$ 'nin benzetimin doğruluk derecesine

etkisi incelenmiş ve optimal yükseklik 1.5 m dolgulu yatak yüksekliği için 3.75 cm olarak bulunmuştur. Benzetim kodu literatürden alınan bazı çalışmalarla, teorik olarak, karşılaştırılmış ve belirli bir geri akış oranında, damıtma derişimlerinin davranışı önceki çalışmalardaki ile aynı bulunmuştur. Benzetim kodu ayrıca farklı geri akış oranları, yoğunlaştırıcıda tutulan farklı sıvı miktarları ve kazana beslenen farklı ısılar için çalıştırılmış ve literatürdeki benzer çalışmalarla karşılaştırılarak, geçerliliği saptanmıştır. Ayrıca, benzetim algoritmasının doğru sonuçlar verdiğini belirlemek için laboratuvar ölçülerinde dolgulu bir damıtma kolonu kullanılarak, etanol ve su karışımını ayırştırmak için deneyler yapılmıştır. Deney sonuçlarının benzetim kod sonuçları ile iyi bir uyum içinde olduğu görülmüştür.

Dinamik modelin doğrulanmasından sonra, ürün miktarını arttırmak için, optimum işletme politikası, kapasite faktör yaklaşımı kullanılarak sayısal olarak araştırılmıştır. Geri dönüşümün kapasite faktör üzerine olan etkisini incelemek için, kolon, atık tanklarında biriken miktarlar geri döndürülerek ve döndürülmeyerek çalıştırılmıştır. Atık tanklarında biriken miktarların geri döndürülerek damıtılmasının ayırma etkisini %28 arttırdığı görülmüştür.

Anahtar Kelimeler: Çok Bileşenli Kesikli Damıtma Sistemi, Dolgulu Kolonların Dinamik Modellemesi, Optimal İşletme

To Mehmet Ali and Hilmiye CEYLAN,



## **ACKNOWLEDGMENTS**

Firstly, I am grateful to my supervisor, Prof. Dr. Canan ÖZGEN, for guiding me not only on my thesis but also on my life to make me a better person by being a model with her leadership, friendship, patience, politeness and wisdom. The days when I saw her smile and felt her encouragement were the most beautiful times of my life.

I would like to thank Almıla BAHAR for her help and giving answers by not disappointing me to my idea, "Almıla knows the answer of this error". I would like to thank M. Oluş ÖZBEK due to his patient on helping me about computer technology and MATLAB software and, Özgecan DERVİŞOĞLU who always encourage and felt me like I was at a stand up show of a star at every meeting when I felt unhappy. I also would like to thank Uğur YILDIZ due to his valuable discussions on batch modeling and simulation algorithms. Also, I would like to thank to my roommates, Hulya TATAS and Nihan KARAKOC due to their valuable friendship.

I am also grateful to my best friends, Cemile KESERCİ and Mustafa SEMİZ. Their encouragement, friendship and moral support are the best motivation for me.

I would like to thank Ferit KOYDEMİR due to his technical, financial and moral support throughout my thesis study, this study would not end without his support.

Financial support of The Scientific and Technological Research Council of Turkey (TUBITAK) is also acknowledged.

Last but certainly not least, I would like to thank my family, who always supported me whatever I do and taught me how to face challenges and were felt near to me even we were apart.

## TABLE OF CONTENTS

|  |      |
|--|------|
| ACKNOWLEDGMENTS .....  | ix   |
| TABLE OF CONTENTS .....  | x    |
| LIST OF TABLES .....   | xiii |
| LIST OF FIGURES .....  | xv   |
| NOMENCLATURE .....   | xvii |
| CHAPTERS   |      |
| 1. INTRODUCTION .....  | 1    |
| 2. LITERATURE SURVEY .....   | 5    |
| 2.1 A Brief Description of a Packed Distillation Column .....            | 5    |
| 2.1.1 Mass Transfer Coefficients .....                                   | 6    |
| 2.1.2 Effective Interfacial Area .....                                   | 9    |
| 2.1.3 Pressure Drop, Loading and Flooding Capacity .....                 | 9    |
| 2.1.4 Holdup .....   | 11   |
| 2.1.5 Flow Maldistribution .....   | 12   |
| 2.2 Design Methods for Packed Columns .....                              | 13   |
| 2.3 Dynamic Modeling of Packed Columns and Solving Model Equations ..... | 13   |
| 2.4 Optimal Operation Policy .....                                       | 17   |
| 3. MULTICOMPONENT BATCH PACKED DISTILLATION COLUMN MODELING .....        | 20   |
| 3.1 Model Assumptions .....  | 20   |
| 3.2 Batch Packed Distillation Column Dynamic Modeling .....              | 22   |
| 3.2.1 Conservation Equations .....                                       | 22   |
| 3.2.2 Determination of Flow Rates at Finite External Reflux Ratio .....  | 28   |
| 3.2.3 Determination of Flow Rates at Total Reflux .....                  | 29   |
| 3.2.4 Determination of Loading and Flooding Points .....                 | 30   |
| 3.2.5 Calculation of Molar Hold up and Pressure Drop .....               | 30   |
| 3.2.6 Predictions of Physical Properties .....                           | 32   |
| 3.2.7 Algebraic Equations and Thermodynamics .....                       | 32   |

|  |    |
|--|----|
| 3.3 Initial Conditions for the Simulation .....  | 34 |
| 3.4 Storage Tank Models .....  | 35 |
| 3.5 Model Equations.....   | 36 |
| 4. OPTIMAL OPERATION OF A MULTICOMPONENT BATCH PACKED DISTILLATION COLUMN.....                 | 38 |
| 4.1 Capacity Factor .....  | 38 |
| 4.2 Problem Statement.....   | 39 |
| 4.3 Optimization Problem Formulation.....  | 39 |
| 5. SIMULATION CODE.....  | 41 |
| 5.1 Main Simulation Code .....   | 43 |
| 5.2 Thermodynamic Library.....   | 44 |
| 5.3 Optimization Functions.....  | 49 |
| 6. CASE COLUMN AND EXPERIMENTAL SET- UP.....   | 51 |
| 6.1 Case Study 1: Hydrocarbons.....  | 51 |
| 6.2 Case Study 2: Polar mixtures .....   | 54 |
| 6.2.1 Experimental Set – Up.....   | 54 |
| 6.2.2 Experimental Procedure.....  | 55 |
| 7. RESULTS AND DISCUSSION .....  | 56 |
| 7.1 Effect of Incremental Bed Height on the Results of the Simulation .....                    | 56 |
| 7.2 Verification of Simulation Code by the Study of Yıldız (2002) .....                        | 57 |
| 7.3 Verification of Simulation Code for Hydrocarbon Mixtures.....                              | 60 |
| 7.4 Experimental Verification of Simulation Code for Polar Mixtures .....                      | 72 |
| 7.5 Determination of Optimum Reflux Ratio .....  | 74 |
| 8. CONCLUSIONS AND RECOMMENDATIONS .....   | 81 |
| REFERENCES .....   | 83 |
| APPENDICES   |    |
| A. MASS TRANSFER AND EFFECTIVE INTERFACIAL AREA CORRELATIONS FOR RANDOM PACKING MATERIALS..... | 88 |
| B. CONSTANTS OF PRESSURE DROP EQUATION FOR DUMPED PACKING MATERIALS ...                        | 91 |
| C. SIMULATION CODE.....  | 92 |
| C.1 Main Program Codes.....  | 92 |
| C.1.1 Glob_Decs.m.....   | 92 |
| C.1.2 Glob_Initial.m.....  | 93 |
| C.1.3 Mass_Hydrocarbons.m .....  | 94 |
| C.1.4 Mass_Polar.m.....  | 97 |
| C.1.5 PressureProfile.m.....   | 98 |
| C.1.6 Plant_File_Packed.m .....  | 99 |

|   |     |
|---|-----|
| C.2 Optimization Code .....                                   | 112 |
| C.2.1 OptimizeR.m .....                                       | 112 |
| C.2.2 Batch_con.m .....                                       | 113 |
| C.2.3 Batch_obj.m.....  | 113 |
| C.3 Thermodynamic Library .....                               | 113 |
| C.3.1 thermo_Init.m .....                                     | 113 |
| C.3.2 thermo_Equilibrium_Hydrocarbons.m.....                  | 113 |
| C.3.3 thermo_Equilibrium_Polar.m .....                        | 114 |
| C.3.4 thermo_Enthalpy.m .....                                 | 114 |
| C.3.5 thermo_L_Density.m.....                                 | 114 |
| C.3.6 thermo_G_Density.m .....                                | 114 |
| C.3.7 thermo_LIBRARY.f .....                                  | 115 |
| C.3.8 thermo_LIBRARY.h .....                                  | 138 |
| C.3.9 common_plant.h.....                                     | 138 |
| C.3.10 parameter.h .....                                      | 139 |
| C.3.11 thermo_data.dat.....                                   | 139 |
| D. STUDIED CHEMICALS .....                                    | 140 |
| E. DETAILS OF EXPERIMENTAL SET UP.....                        | 146 |
| E. 1. Calibration Curve of Peristaltic Pumps .....            | 146 |
| E. 2. Details for the Analysis of Compound Compositions ..... | 147 |

## LIST OF TABLES

|   |    |
|---|----|
| <b>Table 2.1</b> The Volumetric Mass Transfer Coefficient Correlations Used in Some Studies. ....             | 8  |
| <b>Table 2.2</b> Summary of Numerical Methods Used in Different Simulation Studies. ....                      | 16 |
| <b>Table 3.1</b> Important Features of Batch Distillation Column Models. ....                                 | 23 |
| <b>Table 3.2</b> Binary Interaction Parameters, $b_{ij}$ . ....   | 34 |
| <b>Table 3.3</b> Summary of MPDC Model Equations. ....  | 36 |
| <b>Table 5.1</b> The Overall Structure of the Simulation Code. ....   | 42 |
| <b>Table 6.1</b> Column and Feed Specifications. ....   | 53 |
| <b>Table 6.2</b> Additional Specifications for the Packed Distillation Column Used in the<br>Simulation. .... | 53 |
| <b>Table 6.3</b> Specifications of Used Random Packing Material. ....   | 54 |
| <b>Table 7.1</b> Internal Reflux Ratio Profile of the Study of Yıldız (2002). ....                            | 57 |
| <b>Table 7.2</b> Parameters Used in Different Simulation Runs. ....   | 61 |
| <b>Table 7.3</b> Distillate Compositions at Different Reflux Ratios at Two Hours Elapsed Time. ...            | 63 |
| <b>Table 7.4</b> Percentage Changes of Distillate Compositions at Different Internal Reflux Ratios.<br>.....  | 63 |
| <b>Table 7.5</b> Distillate Compositions at Different Height of Packings at Two Hours Elapsed<br>Time. ....   | 64 |
| <b>Table 7.6</b> Percentage Change of Distillate Compositions at Different Height of Packings. ...            | 65 |
| <b>Table 7.7</b> Distillate Compositions at Different Condenser Holdups for Two Hours Elapsed<br>Time. ....   | 66 |
| <b>Table 7.8</b> Percentage Change of Distillate Compositions at Different Condenser Holdups. .               | 67 |
| <b>Table 7.9</b> Distillate Compositions at Different Condenser Holdups for 4.5 Hours Elapsed<br>Time. ....   | 68 |
| <b>Table 7.10</b> Percentage Change of Distillate Compositions at Different Condenser Holdups.<br>.....       | 69 |
| <b>Table 7.11</b> Distillate Compositions at Different Heat Loads for Two Hours Elapsed Time...               | 71 |
| <b>Table 7.12</b> Percentage Change of Distillate Compositions at Different Heat Loads. ....                  | 71 |

|   |     |
|---|-----|
| <b>Table 7.13</b> Experimentally Measured Liquid Ethanol Compositions in Mole Fractions in the Distillate. ....                             | 72  |
| <b>Table 7.14</b> Effect of Rounded Value of Internal Reflux Ratio on Capacity Factor. ....   | 75  |
| <b>Table 7.15</b> List of Some Capacity Factor Values with Rounded Times to Switch another Tank. ....                                       | 75  |
| <b>Table 7.16</b> Results of the Optimization Code to Maximize Distilled Product Amount for Cyclohexane – n-Heptane – Toluene Mixture. .... | 76  |
| <b>Table 7.17</b> Distillate Compositions and Amounts for the Optimum Reflux Ratio Profile. ....  | 77  |
| <b>Table 7.18</b> Simulation Results of the First Recycle Process. ....   | 78  |
| <b>Table 7.19</b> Simulation Results of Third Distillation Phase. ....  | 79  |
| <b>Table 7.20</b> Summary of the Operations with Recycling. ....  | 80  |
| <b>Table A.1</b> Correlations for the Gas And / Or Liquid Side Mass Transfer Coefficients for Random Packings. ....                         | 88  |
| <b>Table A.2</b> Correlations of Effective Interfacial Area for Random Packing Materials. ....  | 90  |
| <b>Table B.1</b> Constants for Some Dumped Packing Materials. ....  | 91  |
| <b>Table D.1</b> Specifications for Hydrocarbons. ....  | 140 |
| <b>Table D.2</b> List of Correlations Used to Predict Physical Properties of the Hydrocarbons. ..   | 141 |
| <b>Table D.3</b> Physical Properties of Ethanol and Water Utilized in the Simulation Runs. ....   | 144 |
| <b>Table D.4</b> Specifications for Ethanol and Water. ....   | 144 |
| <b>Table D.5</b> NRTL Model Parameters. ....  | 145 |
| <b>Table D.6</b> Constants for Antoine Equation. ....   | 145 |
| <b>Table D.7</b> Constants for Wagner Equation. ....  | 145 |
| <b>Table E.1</b> Specifications for GC. ....  | 147 |
| <b>Table E.2</b> Areas Obtained for Prepared Samples. ....  | 147 |
| <b>Table E.3</b> Data for Base Component Factor of Water. ....  | 149 |

## LIST OF FIGURES

|   |    |
|---|----|
| <b>Figure 2.1</b> Loading and Flooding Points for a Packed Column at Total Reflux. ....                           | 10 |
| <b>Figure 3.1</b> Schematic View of Studied Batch Packed Distillation Column. ....                                | 23 |
| <b>Figure 3.2</b> Schematic View of a Differential Section of the Column for Material Transfer... ..              | 24 |
| <b>Figure 3.3</b> Schematic View of a Differential Section of the Column for Energy Transfer. ....                | 25 |
| <b>Figure 4.1</b> STN of the Optimization Problem.....  | 39 |
| <b>Figure 4.2</b> Schematic View of Recycling of the Materials in Slop Cut Tank.....                              | 40 |
| <b>Figure 5.1</b> Simulation Chart of the Batch Packed Distillation Column. ....                                  | 43 |
| <b>Figure 5.2</b> Flow Chart of the Main Simulation Algorithm.....  | 45 |
| <b>Figure 5.3</b> Flowchart of Simulation Algorithm for Optimum Reflux Ratio Profile. ....                        | 50 |
| <b>Figure 6.1</b> Residue Curve Map for Cyclohexane – n - Heptane – Toluene at 1 atm.....                         | 52 |
| (Jiménez et al., 2002). ....  | 52 |
| <b>Figure 6.2</b> Schematic View of Experimental Set – Up.....  | 55 |
| <b>Figure 7.1</b> Comparison of Distillate Compositions for Different Incremental Elements.....                   | 57 |
| <b>Figure 7.2</b> Comparison of Distillate Compositions at Total Reflux. ....                                     | 58 |
| <b>Figure 7.3</b> Comparison of Distillate Compositions at Internal Reflux Ratio of 0.875.....                    | 59 |
| <b>Figure 7.4</b> Comparison of Distillate Compositions with a Specified Reflux Ratio Profile.....                | 60 |
| <b>Figure 7.5</b> Effect of Internal Reflux Ratio on the Distillate Compositions. ....                            | 62 |
| <b>Figure 7.6</b> Effect of Height of Packings on the Distillate Compositions. ....                               | 64 |
| <b>Figure 7.7</b> Effect of Condenser Holdups on the Distillate Compositions.....                                 | 66 |
| <b>Figure 7.8</b> Effect of Condenser Holdup on the Distillate Compositions with One Hour<br>Startup Period. .... | 68 |
| <b>Figure 7.9</b> Effect of Condenser Holdup on Distillate Compositions with 2.5 Hours Startup<br>Period. ....    | 69 |
| <b>Figure 7.10</b> Effect of Heat Load on Distillate Compositions. ....   | 70 |
| <b>Figure 7.11</b> Comparison of Results Obtained By Simulation and Experiments.....                              | 74 |
| <b>Figure 7.12</b> The Liquid Distillate Compositions Obtained under Optimum Reflux Ratio<br>Profile. ....        | 76 |

|  |     |
|--|-----|
| <b>Figure 7.13</b> Schematic View of Recycle Process for the Proceeding Batch Operation..... | 78  |
| <b>Figure E.1</b> Calibration Curve of the Peristaltic Pumps.....                            | 146 |
| <b>Figure E.2</b> Behavior of Base Component of Water.....                                   | 149 |



## NOMENCLATURE

|                 |  |
|-----------------|--|
| A               | Tower cross-sectional area [ $\text{m}^2$ ]  |
| $A_1, A_2, A_3$ | Constants for Antoine equation   |
| $a_e$           | Effective interfacial area per unit volume of packing volume [ $1/\text{m}$ ]  |
| $a_p$           | Specific packing surface area per unit volume of packing volume [ $1/\text{m}$ ]   |
| b               | Binary interaction parameter   |
| B               | Constant in Equation D.6   |
| $\Delta B$      | Group contribution number in Equation D.10   |
| $C_1, C_2, C_3$ | Packing constants for pressure drop correlations   |
| c               | Packing specific constant = 2 if $d_p < 15$ mm,<br>otherwise $c = 5.23$ in Equation 3.12 (Djebbar et al., 2002)                                      |
| c               | Constant in Equations 3.56 and 3.57  |
| D               | Distillate flow rate [ $\text{mol}/\text{hr}$ ]  |
| D               | Diffusion coefficient [ $\text{m}^2/\text{s}$ ]  |
| $d_p$           | Particle diameter [ $\text{m}$ ]   |
| $f_o$           | Friction factor for flow past a single particle  |
| f               | Fugacity, pure species j   |
| Fr              | Froude number  |
| g               | Gravitational constant [ $\text{m}/\text{s}^2$ ],<br>Interaction energy [ $\text{Pa} \cdot \text{m}^3 / \text{mol}$ ] in Equation 3.66               |
| G               | Specific Gibbs free energy [ $\text{J}/\text{mol}$ ]   |
| h               | Liquid mixture enthalpy [ $\text{J}/\text{mol}$ ]  |
| H               | Vapor mixture enthalpy [ $\text{J}/\text{mol}$ ]   |
| $h_o$           | Liquid holdup below the loading point [ $\text{m}^3/\text{m}^3$ ]  |
| $h_L$           | Liquid holdup in a packed bed above the loading point [ $\text{m}^3/\text{m}^3$ ]  |
| k               | Mass transfer coefficient [ $\text{mol}/(\text{h} \cdot \text{m}^2 \cdot \text{mol frac})$ ] and, in Equations 3.12 and 3.19 [ $\text{m}/\text{s}$ ] |
| K               | Overall mass transfer coefficient [ $\text{mol}/(\text{h} \cdot \text{m}^2 \cdot \text{mol frac})$ ]   |

|                                   |  |
|-----------------------------------|--|
| L                                 | Liquid flow rate [mol/hr]  |
| M                                 | Molar holdup [mol]   |
| M'                                | Molar holdup / height [mol/m]  |
| m                                 | Slope of the equilibrium line  |
| Mw                                | Molecular weight [kg/mol], in Equation D.1 and D.18 [g/mol]                        |
| N                                 | Constant in Equation D.1, given by Equations D.2 and D.3                           |
| N                                 | Mass flux in Equation 3.1 [mol/(h.m <sup>2</sup> )]                                |
| N <sup>o</sup>                    | Constant in Equation D.7   |
| P                                 | Pressure [Pa], in Equation D.18 [psia], in Equation 7.1 [mmHg]                     |
| P                                 | Product tank   |
| Q                                 | Interaction parameter in Equation D.4  |
| Q <sub>p</sub>                    | Heat loss per unit length [J/(h. m)]   |
| Q                                 | Heat load [J/hr]   |
| R                                 | External reflux ratio (L/D)  |
| R <sub>g</sub>                    | Ideal gas constant [Pa.m <sup>3</sup> /(mol.K)]                                    |
| R <sub>p</sub>                    | Internal reflux ratio (L/V)  |
| Re                                | Reynolds Number  |
| S                                 | Slop cut tank  |
| Sc                                | Schmidt Number   |
| t                                 | Time [hr]  |
| T                                 | Temperature [K], in Equation D.18 [°R]   |
| T <sub>o</sub>                    | Constant in Equation D.6   |
| u                                 | Superficial velocity through a packed bed [m/s]                                    |
| U                                 | Overall heat transfer coefficient [J/(h. K)]                                       |
| W <sub>1</sub> ... W <sub>4</sub> | Constants for Wagner equation  |
| We                                | Weber number   |
| x                                 | Liquid fraction [mol/mol]  |
| x*                                | Equilibrium liquid mole fraction [mol/mol]   |
| y                                 | Vapor fraction [mol/mol]   |
| y*                                | Equilibrium vapor mole fraction [mol/mol]  |
| z                                 | Packing height at any time [m]   |
| Z                                 | Total packing height [m]   |
| V                                 | Vapor flow rate [mol/hr]   |
| V                                 | Molar volume at the normal boiling point [cm <sup>3</sup> /gmole] in Equation D.18 |

**Greek Letters:**

|          |                                 |
|----------|---------------------------------|
| $\alpha$ | Binary parameter for NRTL model |
| $\beta$  | Base component factor           |

|               |   |
|---------------|---|
| $\Delta$      | Increment   |
| $\varepsilon$ | Bed void fraction (porosity) [ $\text{m}^3/\text{m}^3$ ]  |
| $\gamma$      | Activity coefficient, species j in solution   |
| $\lambda$     | Latent heat of vaporization [ $\text{J}/\text{mol}$ ]   |
| $\kappa$      | Area of the components obtained from GC   |
| $\mu$         | Viscosity [ $\text{kg}/(\text{m}\cdot\text{s})$ ]   |
| $\phi$        | Fugacity coefficient  |
| $\rho$        | Density [ $\text{kg}\cdot\text{m}^{-3}$ ], in Equations D.16 and D.17 [ $\text{kmole}/\text{m}^3$ ] |
| $\sigma$      | Surface tension [ $\text{mN}/\text{m}$ ]  |
| $\tau$        | Binary parameter for NRTL model   |

**Subscripts:**

|     |                                   |
|-----|-----------------------------------|
| B   | Reboiler                          |
| br  | Reduced normal boiling            |
| c   | Critical                          |
| dry | Dry                               |
| f   | Flooding                          |
| i   | Discretization level number index |
| irr | Irrigated                         |
| j   | Component number index            |
| k   | Stage index                       |
| L   | Liquid                            |
| m   | Mixture                           |
| r   | Reduced                           |
| T   | Total                             |
| V   | Vapor                             |

**Superscripts:**

|      |   |
|------|---|
| 1    | State 1 in Figure 4.1                           |
| 2    | State 2 in Figure 4.1                           |
| **   | Desired purity level of component j             |
| o    | Initial   |
| feed | Fresh feed at the start of the system operation |
| L    | Liquid  |
| V    | Vapor   |

**Abbreviations:**

|      |                      |
|------|----------------------|
| CAP  | Capacity Factor      |
| DL   | Discretization Level |
| EtOH | Ethanol              |

|        |  |
|--------|--|
| GC     | Gas Chromatography                               |
| HETP   | Height Equivalent to a Theoretical Plate         |
| MBPDC  | Multicomponent Batch Packed Distillation Column  |
| NC     | Number of Components                             |
| NRTL   | Non Random Two Liquid                            |
| NT     | Number of Differential Element of Packed Section |
| ODE    | Ordinary Differential Equation                   |
| PDE    | Partial Differential Equation                    |
| PR-EOS | Peng Robinson Equation Of State                  |
| SA     | Simulated Annealing                              |
| SQP    | Sequential Quadratic Programming                 |
| STN    | State Task Network                               |
| VLE    | Vapor Liquid Equilibrium                         |

**Special Notation:**

circumflex^  
(as in  $\hat{f}_j$ ) Denotes property of a component in a mixture

# **CHAPTER 1**

## **INTRODUCTION**

The separation operation called as distillation, which uses “gas and liquid phases at essentially the same temperature and pressure for the coexisting zones”, is one of the most commonly used separation processes in chemical and petroleum industries (Perry et al., 1997).

Among the two different types of distillation columns; plate and packed, due to the increase of contact area between liquid and gas phase, which results in high efficiency, high capacity to resist to corrosive materials, low pressure drop and small liquid holdup, packed columns are preferred compared to plate columns in design applications (Perry et al., 1997).

Packed columns can be filled with either structured (ordered) or random (dumped) packing materials. Although structured packing materials are used commonly in the design of distillation columns, as a result of the new generations, the random packing materials are still used with an increase demand, due to the knowledge of the behavior of random packing materials on mass transfer and interfacial area. Hence, this study is mainly focused on packed distillation columns with random packing materials.

Random packing materials like Berl saddle, Raschig and Pall rings can be made of different type of materials: carbon, glass, metal, etc. Difference on surface texture and material diameter results in changes in the characteristics of the packing like porosity and interfacial area. Therefore, during design process, those characteristics must be taken into consideration in detail.

The choice of the type of operation of distillation, as batch or continuous depends on the feed amount and on the characteristics of the feed components. Nowadays, batch distillation is more commonly used due to its convenience for low volume fine chemicals and biochemicals (Betlem, 2000) and its advantages like "flexibility, high product purity and possibility of multiple fraction operation" (Li et al., 1998). The "flexibility" feature, which provides to design different configurations of the column easily, can be utilized to overcome uncertainties in product specification. Moreover, in batch distillation; separation of  $n$  components can be handled in a single column by using product and slop cut tanks. However, in continuous distillation, to separate  $n$  components,  $n - 1$  continuous distillation column must be used (Diwekar, 1996).

There are some disadvantages, beside advantages, related to the use of batch distillation columns, such as, continuous change of feed stocks and complex column dynamics. The complexity of column dynamics increases as the number of component in the feed mixture increases from binary to multicomponent. These disadvantages reduce batch distillation column's competitiveness significantly and require research on optimum operating reflux ratios. It is very difficult to determine the composition changes with time throughout the batch distillation columns at optimum reflux ratios by performing experiments or by taking data from the column due to complex column dynamics (Kreul et al., 1999). Fieg et al. (1994) stated that, the dynamic behavior of an actual distillation column can be predicted at different operating conditions by using the simulation algorithm and any malfunction of the system can be controlled without making any unnecessary investment. Hence, preparation of simulation algorithm and determination of optimum reflux ratio profile to maximize amounts of products for a multicomponent batch packed distillation column are essential.

There are two aims of this study. The primary objective is to develop a dynamic model for the simulation of a batch packed distillation column with random packing material for multicomponent mixtures at low operating pressures. Thus, both theoretical and experimental studies are carried out. The dynamic model is a set of equations, which rely mainly on mass and energy transfer mechanisms and thermodynamic equations. A simulation algorithm is prepared to solve the dynamic model with the aid of MATLAB programming language (Version 7.0.1. 24704 (R14)). It is desired to obtain composition, temperature and flow rate profiles for reflux – drum condenser, reboiler and packed sections during the operation with given column specifications like diameter of the column, type of the packing material, porosity and height of the packed section with the help of the simulation algorithm. Also, it is aimed to obtain the irrigated and flooding pressure drop and the value of design gas rate at specified operating conditions. Verification of simulation code

is conducted for hydrocarbon and polar mixtures. For the former mixtures, simulation code is tested by comparing product compositions by nine different simulation runs, while for latter mixtures experimental studies are carried by using ethanol and water mixture at a packed distillation column with 0.4 m height, 5 cm diameter and filled with 5 mm glass Raschig rings.

Second objective of this study is to determine the optimum reflux ratio profile of the multicomponent batch packed distillation column system, to maximize the amount of distillate of a specified concentration for a given time. Optimization is done by using capacity factor (CAP) approach which is a function of time and composition. The optimum reflux ratio profile and switching times between product cut and slop cut tanks will be found by utilizing composition of the reflux - drum – condenser part and the knowledge of desired purities of the main products. Also, the effect of recycling of molar holdups of slop cut tanks for the next batch operation to maximize the amount of distilled products at a packed distillation column is analyzed by finding optimum reflux ratio profile in the context for hydrocarbon mixtures.

Among the eight chapters of the thesis, in Chapter 2, literature survey related to the work is explained with a description of packed columns together with the discussion of the important effect of type of random packing materials on gas phase mass transfer. Also, the parameters related to hydrodynamics and mass transfer are explained and their experimental or theoretical determinations are discussed. A discussion about design methods to prepare model equations of a packed distillation column and also about numerical approaches used by previous studies is given. Moreover, research on optimum reflux ratios is discussed. In Chapter 3, modeling studies for a multicomponent batch packed distillation with random packing material are given with model assumptions considering literature with correlations related to pressure drop and physical properties with their percentage errors. Lastly, storage tank models are discussed. In Chapter 4, optimal operation conditions are discussed to maximize distilled product amount for specified concentrations and the optimization problem is identified and formulated. Simulation code and its working principle are explained in detail in Chapter 5. Experimental setup and procedure are explained in Chapter 6. Results and discussions are given with figures and tables, in Chapter 7. The studies on the verification of simulation code are started with comparison of findings with these of the study of Yıldız (2002). After that, the test of the model for various cases by using cyclohexane – toluene – n-heptane mixture is described and then, the experimental verification of simulation code is given for ethanol and water mixture. Moreover, the optimum reflux ratio profile, related to

capacity factor values and recycle of molar holdups of slop cut tanks are discussed in detail. In the last chapter, Chapter 8, conclusions and recommendations for further work are given.



## CHAPTER 2

### LITERATURE SURVEY

In this chapter, the literature survey for a packed distillation column is given in detail with a brief description of a packed distillation column system, advantages, and hydrodynamic and mass transfer issues. Also, design methods for packed columns which are found in literature are given with a literature survey about dynamic modeling of packed columns. Previous studies on optimal operation policy of a batch distillation are also given.

#### 2.1 A Brief Description of a Packed Distillation Column

Packed columns are used in distillation, which is a commonly used unit operation in process industries, as well as in absorption and stripping applications (Geankoplis, 2003).

There are several cases when the use of packed columns is preferred to the use of tray columns. Some of them are stated as (Perry et al., 1997):

- If alloy – metal packing materials are not necessary, the cost of packed column is usually less than the cost of the plate column when tower diameter less than 0.6m,
- Using resistant packing materials, caustic mixtures can be operated in packed columns,
- Liquid holdup in packed column will be much more less than that in plate column,
- Packings have a characteristic of required efficiency for low pressure drop,
- In packed columns, liquid agitation by gas occurs lesser than the plate columns.

There are two different types of packing arrangement; random and structured. Raschig rings and Berl saddles are known as first generation packing materials whereas Pall ring is known as second generation packing material, where packing materials are placed in the column in

a random arrangement, usually by dumping. In structured packing such as Sulzer and Flexipac, carefully installed elements are designed specifically to fit the column dimensions. The efficiency of packing materials displays variations due to difference in void fraction and effective surface area. Macias – Salinas and Fair (2000) studied on the axial mixing in four packing materials: 25.4 mm ceramic Raschig rings, 25.4 mm metal Pall rings, SulzerBX structured packing and Flexipac2 structured packing. Tracer experiments were made in a large scale packed column (0.43 m diameter) by changing the liquid (water) flow rates from 3.25 to 8.5 kg / m<sup>2</sup>sec and gas (air) flow rates from 0.25 kg/m<sup>2</sup>sec to flooding point. It is found that “axial mixing in the gas increases with both gas and liquid rates, whereas liquid – phase axial mixing is a decreasing function of liquid rate and is sensitive to gas rate up to the flooding point” for first – generation random packing materials. The best packing for the axial mixing in gas phase is found to be Raschig rings whereas in the liquid phase as Flexipac2.

Putting packing material into column brings some other considerations into account as hydraulics and mass transfer. The issues of hydrodynamics and mass transfer are volumetric mass transfer coefficients for liquid and gas phase, effective interfacial area, pressure drop, loading and flooding capacity, total liquid holdup and flow maldistribution. Extensive research has been done on the issues of packing material since 1930s. Many experiments have been performed and correlations have been derived.

### **2.1.1 Mass Transfer Coefficients**

Mass transfer coefficients for liquid and gas phases are represented by  $k_L$  and  $k_V$ , respectively. Estimation of mass transfer coefficients for packed distillation columns can be done by making experiments or by using correlations for film coefficients or by predicting mass transfer film coefficients (Geankoplis, 2003). Overall mass transfer coefficients can be measured experimentally easily. However, to determine the individual mass transfer coefficients,  $k_L a$  and  $k_V a$  experiments must be designed carefully. Another way to estimate mass transfer coefficients is to use correlations for film coefficients. There are numerous correlations developed by several authors since 1940s. The correlations for mass transfer coefficients are empirical and they can show deviations up to 25%. A summary of the mass transfer correlations for random packing materials are given in Table A.1 (Appendix A) (Wang et al., 2005). The third way of predicting mass transfer coefficients is by the use of Equations 2.1 and 2.2 (Geankoplis, 2003):

$$H_Y = \left( \frac{0.226}{f_p} \right) \left( \frac{Sc}{0.660} \right)^{0.5} \left( \frac{G_x}{6.782} \right)^{-0.5} \left( \frac{G_y}{0.678} \right)^{0.35} \quad (2.1)$$

$$H_x = \left( \frac{0.357}{f_p} \right) \left( \frac{Sc}{372} \right)^{0.5} \left( \frac{G_x / \mu}{6.782 / 0.8937 * 10^{-3}} \right) \quad (2.2)$$

where  $H_Y$  and  $H_x$  are film coefficients for gas and liquid respectively [m],  $f_p$  is relative mass transfer coefficient [-],  $Sc$  is Schmidt number [-] and  $G_Y$  and  $G_x$  are mass flow rates for gas and liquid respectively [kg/(m.s<sup>2</sup>)].

If the table of correlations given in Appendix A is analyzed, it is seen that the individual mass transfer coefficients  $k_L$  and  $k_V$  depend generally on Schmidt number( $Sc$ ), liquid and gas mass velocities, size and shape of the packing material. The correlations for volumetric mass transfer coefficients,  $k_L$  and  $k_V$ , are given by Wang et al. (2005).

The overall mass transfer coefficient correlations that are used in some studies on packed distillation columns are given in Table 2.1.

Bravo et al. (1992) investigated the behavior of mass transfer coefficients for different packing types and different vapor loading points. It is seen that volumetric mass transfer coefficients are affected by rates of vapor and liquid flows. Mass transfer coefficients increase with an increase in vapor velocity at constant liquid flow rates. If liquid flow rate increases then, the efficiency for random packing materials in distillation column will be higher. Moreover, it is demonstrated that the mass transfer coefficient changes sign depending on the packing type and vapor loading. Therefore, to apply any control theory to packed distillation columns the knowledge of the sign of mass transfer coefficient is important.

Wagner et al. (1997) developed a model on mass transfer rates by using the formulas that is derived by Stichlmair et al. (1989) for pressure drop and liquid holdup at packed columns. The study is done on new random packing materials like IMTP, Fleximax and Nutter. The results of the model were in a good agreement with the experimental data.

**Table 2.1** The Volumetric Mass Transfer Coefficient Correlations Used in Some Studies.

| Author(s) – Year            | Used Packing Material           | Used Correlation(s)  |
|-----------------------------|---------------------------------|--|
| Patwardhan and Edgar - 1993 | Sulzer BX                       | $K_y^o a_m = p(10^{nV/A})$ (2.3)   |
| Karacan et al. - 1998       | Raschig ring<br>(20 – 15 mm ID) | $K_{OG} a = b(V_G)^m (V_L)^n$ (2.4)  |
|                             |                                 | $\frac{k'_v S}{D_v} = 0.054 \text{Re}_v^{0.8} \text{Sc}_v^{0.33}$ (2.5)            |
| Attarakih et al. - 2001     | Gempak 2A                       | $K_v a_e = K'_v a_e \left( \frac{P}{RgMT} \right) A_c$ (2.6)                       |
|                             |                                 | $K'_v = \frac{1}{\frac{1}{K'_v} + \frac{m' U_{LS}}{R U_{VS}} \frac{1}{K_L}}$ (2.7) |

Piché et al. (2001a) presented a detailed study using neural network models for the hydraulics and mass transfer in randomly packed distillation columns. "A conceptual fully predictive mechanistic model" was built up by using 10,750 data covering the 1933 - 2000 literature. The model compared with well - known correlations like Billet and Schultes (1991) model, Onda mass transfer correlation (Wang et al., 2005), etc. It is found that, model fits the correlations well and it is stated that without knowing packing constant,  $c$ , the model can be used to determine mass transfer coefficients for random packing materials. Otherwise, it is advised that, Onda mass transfer correlation (Wang et al., 2005) can be used to determine mass transfer coefficient if packing constant,  $c$  is known.

Actually, Onda mass transfer correlation (Wang et al., 2005) is widely used in stripping applications. However, Wang et al. (2005) stated that, this correlation can be used to determine mass transfer coefficients for absorption, stripping, and distillation applications. Random packing materials which are first generation packing materials, has many investigations over the years, and to find information about them is not too difficult. Since, the packing constants are generally known for random packing materials, in this study Onda mass transfer correlation (Wang et al., 2005) will be used to predict mass transfer coefficients and effective interfacial area.

### **2.1.2 Effective Interfacial Area**

As in the mass transfer coefficients for gas and liquid phases, there is also an extensive literature on effective interfacial area of random and structured packing materials. Physical methods i.e. electroresistivity, light transmission and mass transfer measurements are the ways of determining the effective interfacial area. However, many researchers chose the mass transfer with a fast chemical reaction to determine effective interfacial area in distillation, stripping and absorption applications.

Wen et al. (2001) studied the 3D profile of the liquid flow that can move horizontally or vertically in a randomly packed distillation column by developing geometry based model. It is demonstrated that two of the flow types, film flow, dripping flow are the main mechanisms of packed columns.

Depending on the findings of Wen et al. (2001) and the previous works, it can be observed that, the definition of interfacial area differs from one publication to another. The most important interfacial area is the wetted interfacial area where mass transfer occurs. This area may be less than the actual interfacial area. The difference between effective interfacial area and wetted interfacial area is that, effective area includes drippings, gas bubbles and jets while wetted interfacial area does not (Perry et al., 1997). Table A.2 (Appendix A) tabulates effective interfacial area correlations with their investigators for random packing materials.

Since Onda effective interfacial area correlation (Wang et al., 2005), which is the most reliable predictive equation for packed distillation column design, is studied comprehensively considering on hydrodynamic and liquid physical properties on the wetted surface area of random packing materials (Wang et al., 2005).

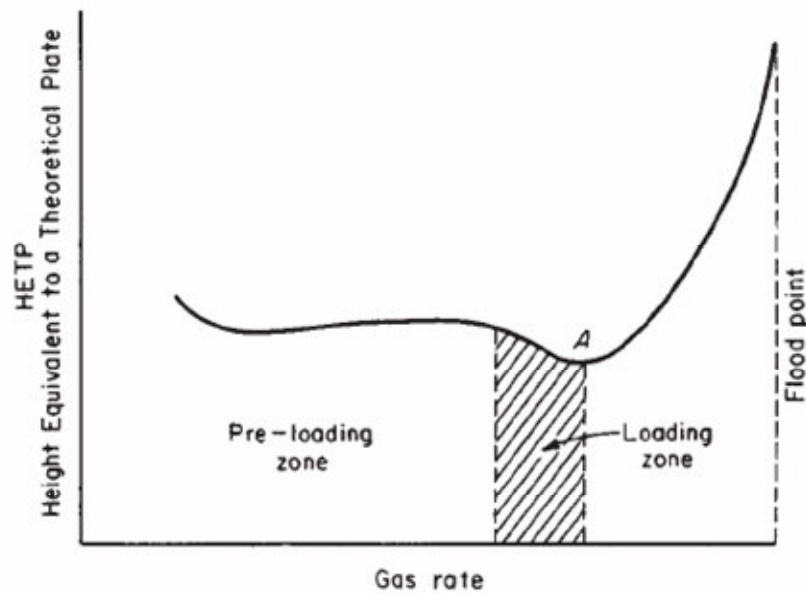
### **2.1.3 Pressure Drop, Loading and Flooding Capacity**

Pressure drop in a packed distillation column is affected primarily by the followings:

- fluid flow rates,
- density and viscosity of the fluids,
- size, shape and interfacial area of the packing materials.

Pressure drop at a packed distillation column can be calculated by using several methods, but in the most widely used method correlations that are based on experimental data which acquired countercurrent flow of water and air for different packing materials are used.

In estimation of pressure drop, there are two important points that must be taken into consideration carefully; loading and flooding points. Loading point is where liquid starts to fill up the column and is reducing the effective space for gas flow. On the other hand, flooding point is where a liquid layer develops on the top of the packed section. Flooding point is the upper boundary condition of the pressure drop. These points can be visualized in Figure 2.1 (Perry et al., 1997).



**Figure 2.1** Loading and Flooding Points for a Packed Column at Total Reflux.

In the modeling of hydrodynamics of a packed column, either channel model or particle model can be used. In the channel model, it is assumed that there are several small channels for the flow of fluids throughout the packing materials. As gas flows upward through the channels, the liquid flows down through the walls of the channels. The flow of the liquid decreases the area for the flow of gas which causes the pressure drop. On the other hand, in the particle model, gas flows around the packing particles instead of channels. The presence of liquid flow reduces the void space between the packing materials.

Stichlmair et al. (1989) derived a pressure drop correlation for countercurrent gas – liquid packed columns by using the particle model as in the case of Ergun equation. Friction factor, bed porosity and operating liquid holdup are taken into consideration in developing correlations for irrigated and flooding pressure drops. It is also shown that the liquid holdup below loading point is a function of Froude number and the obtained relationship is valid for liquid viscosities up to 5 centipoises.

Piché et al. (2001b) developed a flooding capacity correlation by utilizing artificial neural network modeling. Experimental data in literature is used to validate the correlation. The correlation shows 16.1% average relative error with standard deviation 20.6%. However, this correlation can be used for a wide range of random packing materials since an extensive experimental data from literature is used. Also, by this correlation it is shown that flooding capacity depends on “liquid superficial velocity, liquid viscosity, gas density, bed porosity, packing surface area and the column diameter”.

#### **2.1.4 Holdup**

Another issue in hydrodynamics is holdup. In packed distillation columns, holdups for liquid and/or vapor phase can be considered. The holdup of liquid can be taken as static, operating and total. Static holdup is known as the amount of liquid that remains on the packing material after the liquid is drained from the system. Operating liquid holdup is the amount of liquid that accumulated in the system when the system operates under dynamic conditions. The total holdup is the sum of the static and the operating holdups (Perry et al., 1997). In addition, the amount of total holdup depends on the operation type, whether the system is plate or packed distillation column. Fieg et al. (1994) determined and compared the dynamic behavior of the plate and packed distillation column after disturbances for the development of optimal strategy at product changeovers. It is stated that the type of distillation column has an effect on hold-ups, and packed distillation column has lesser hold-up than the plate distillation column, although same diameter and same column efficiency are used in the distillation columns. On the other hand, the holdup of vapor is generally considered as negligible at low pressures. Choe and Luyben (1987) investigated the importance of assumptions on negligible vapor holdup for distillation columns at low and high pressures. It is stated that assumption of negligible vapor holdup is valid for systems at moderate pressures (up to 10 atm). In high pressure systems, negligible vapor holdup gives 19% error in the time constant.

The amount of liquid that accumulates in the system under dynamic conditions mainly depends on the voidage fraction of packing material and superficial gas velocity. Actually, a detailed study on voidage fraction of the packing material is made by Klerk (2003). They investigated the voidage variation for small column by particle diameter / bed diameter ratios for equal sized sphere particles. Experiments were made by using nonporous glass beads of 0.003m diameter. Experimental results showed that the behavior of the bed voidage oscillates, reaches a constant value for higher column to particle ratios and there are multiple stable values for bed voidage. Although this is the case, for rigorous modeling studies the variation of voidage is neglected. On the other hand, the dependence of holdup on superficial gas velocity is examined by Taiwo and Fasesan (2004). The model is derived from the momentum balance of the liquid film, and demonstrates about 3.5% error with respect to experimental data obtained by using acetone / methanol / ethanol mixtures for different liquid and gas flow rates. It is observed that dynamic liquid holdup decreases with by an increase in the packing height and can be more for systems whose components have higher relative volatility. Moreover, the dynamic liquid holdup increases with an increase in gas velocity of the system components. As the gas velocity increases, the resistance to liquid flow through packing material increases, and then the liquid accumulates at the void spaces of the packed column.

### **2.1.5 Flow Maldistribution**

As stated before, there are five important issues for the hydraulics and mass transfer taking place in packed distillation columns. The last and one of the most important issues is the flow maldistribution. The influence of flow maldistribution has been investigated over the years by several researchers.

As stated by Kister and Braun (1989), the maldistribution can have three affects: pinching effect, lateral mixing effect and uneven flow of liquid through the packed section which reduces column efficiency. Pinching effect is the less delivery of liquid in some regions compared to the other parts of the column. This reduces the ratio of the liquid / gas flow rate, and causes a pinch point in the column. Lateral mixing effect is the movement of liquid and gas horizontally due to maldistribution. The liquid flows down from the liquid distributor if the top of the packed section is not irrigated by liquid. Then, liquid tends to move along the walls of the column. This is known as uneven flow of liquid through the packed section.

Flow maldistribution is a function of packing and tower diameters, and type of the packing. Flow maldistribution does not depend on viscosity and density of liquid. The best liquid



distribution throughout the column can be obtained when ratio of tower diameter / particle diameter is equal to 10. In operation, the effect of lateral mixing is more important than the pinching effect and flow maldistribution can be handled without a serious efficiency loss. However, this is not the case at higher ratio of tower diameter / particle diameter (>40) (Kister and Braun, 1989). In addition, columns with random packing materials gave less flow maldistribution than columns with structured packing materials (Hanley, B. (1999)).

## **2.2 Design Methods for Packed Columns**

A packed distillation column can be designed either by considering packing as a continuous media or by using transfer units. Gorak and Vogelpohl (1985) have showed that use of Height Equivalent to a Theoretical Plate (HETP) in the design of packed distillation column causes very poor results. After that, Krishnamurthy and Taylor (1985) found an answer to the question arisen "as to how variations between component values of quantities such as stage – efficiency, HETP, and HTU can be taken into consideration in the simulation and design of a multistage or packed column" by considering packing as a continuous media in the design of packed column. This method has been used by several authors like Karlström et al. (1992), Pathwardhan and Edgar (1993), Karacan et al. (1998), Mori et al. (1999), Attarakih et al. (2001) and Repke et al. (2004).

## **2.3 Dynamic Modeling of Packed Columns and Solving Model Equations**

In the modeling of a multicomponent batch packed distillation column, mass, energy and momentum balance equations and correlations for liquid and gas phase using ordinary or partial differential equations must be used. The system is very complex. Therefore, simplification of model must be done by making some assumptions.

Skogestad and Morari (1988) explained in detail the behavior of the change of compositions at distillation columns for different cases especially with the existence of internal and external flows. In his study, assumptions, dynamic modeling equations and control studies for the cases are given. It is concluded that big reflux ratios must be applied to the distillation columns when the high purity columns are taken into consideration. The compositions in the column are affected from external flows much more than the internal flows.

Karlström et al. (1992) presented transport phenomena and necessary thermodynamic equations in multicomponent packed bed distillation column. "One-film theory" is used due

to lack of heat accumulation in the main body of the liquid. Therefore, vapor phase is taken as controlling both heat and mass transfer. To simplify the model, equations are written by taking into account radial change only. The nonlinear model equations are also included when the simplified equations are likely to fail. In the model, assuming negligible diffusion interactions between components, "effective diffusivity" method is used. Following this study Gorak (1992) argues that, the use of effective diffusivity can be recommended if the diffusivities of binary subsystems are almost equal and the use of Stefan-Maxwell equations for multicomponent mass transfer is much more applicable than the "effective diffusivity" method.

There are two approaches that have been proposed for solving model equations as Krishnamurthy and Taylor (1985) recommended. In the first approach, packed section is divided into sections and each section is treated as a tray. In the second approach, transfer equations are written for a small incremental section and then balances are integrated for the whole column.

Tommasi and Rice (1970) investigated dynamics of a random batch packed distillation when a step change is given to the reflux ratio. In the design of packed column the approach by considering packing as a continuous media is chosen. Experiments were carried out to validate the model by using ethanol and water mixture. A solution is developed by using nonlinear least - squares method to obtain the behavior of the liquid concentration after the step change given to the reflux ratio. The solution is given in exponential form and the constants in the equation depend on the components of the mixture. Moreover, it is found that parameters that are feature of the final steady state affect the time constants only. Any change of the step disturbance does not have any effect on time constants.

Krishnamurthy and Taylor (1985) solved model equations by using Newton's method and it is observed that there is good agreement between experimental results and simulation.

Some models can include a set of partial differential equations. To solve these equations, there are numerous techniques. Method of orthogonal collocation is one of used methods in the previous studies. However, Krishnamurthy and Taylor (1985) stated that:

*"Cho and Joseph (1983b) found the collocation method to work well for both steady state and dynamic simulation of relatively ideal systems (equilibrium stage model equations and stage efficiency were used to represent the process). However, when nonlinear equilibrium relationships were used along with vapor phase controlled heat and mass transfer rate equations to model packed-bed separation processes (the*

*packed distillation column problem of Von Rosenberg and Hadi (1980) and an absorption problem from Treybal, 1969), the steady state solution could not be directly obtained (Srivastava and Joseph, 1984)."*

Wajge et al. (1997) compared different numerical methods for the simulation of multicomponent batch distillation in packed beds. The use of finite difference method and orthogonal collocation method together for systems that requires higher accuracy is much more applicable than the use of only orthogonal collocation method. In case of using only orthogonal collocation method, higher order orthogonal polynomials must be used which increase computation time. Moreover, it is stated that, if the number of equations for simulation algorithm increases, necessary time to compute the behavior of the system increases.

Karacan et al. (1998) presented the steady state and dynamic behavior of randomly packed distillation column with a thermo siphon reboiler to distill the binary, methanol – water mixture. Partial differential equations are used to model the packed column. In the model equations, the back mixing model is considered basing on two film theory of mass transfer. They are solved by orthogonal collocation on finite elements employing Legendre polynomials due to its power and speed. It is shown that use of orthogonal collocation method results in good accuracy for the system at steady state. However, this result can not be obtained for the unsteady state case.

Mori et al. (1999) presented a simulation algorithm for a packed distillation column at total reflux by means of a rate – based model. Experimental results are consistent with the simulation results. However, the simulation is unsuccessful for evaluation of the effective interfacial area and liquid hold-up distribution, so necessary improvements must be done for a better simulation.

Attarakih et al. (2001) investigated dynamics of a packed distillation system used to separate glycerol and water mixture. The studies on modeling and simulation are made and compared experimentally. The method used before by Patwardhan and Edgar (1993) is used to design packed distillation column. VLE concentrations are calculated by using the UNIFAC model. Dynamic modeling is done for three cases: a) with liquid, vapor and reflux drum holdups, b) liquid and reflux drum holdups and c) with reflux drum holdup only. MATLAB subroutine ode15s (Gear's method) is used for 1<sup>st</sup> and 2<sup>nd</sup> dynamic models due to high stiffness. In the third model Runge – Kutta – Merson method is used to solve set of equations due to existence of nonlinear equations. It is seen that, the Runge – Kutta – Merson method in the

third model requires more computation time and is less accurate than the other models, since there are more steps to compute which causes the growing of "round off and truncation errors".

Jiménez et al. (2002) investigated the nonlinear dynamic modeling of a structured packed distillation column. The experimental data of Nad and Spiegel (1987) is compared with the simulation results, which are conducted for a batch packed column with 167 mm diameter, 8 m height and structured packing material by using BATCHFRAC™, CHEMCAD BATCH™ and HYSYS.Plant® programming packages. It is stated that the use of HYSYS.Plant® is much more efficient than the other software packages. To predict vapor liquid equilibrium compositions, UNIQUAC model is used. Necessary coefficients for this model are taken from the database of Aspen PLUS®.

Repke et al. (2004) worked on the simulation of nonequilibrium model. They conducted experimental runs for three-phase distillation in a structured packed column. The system of equations is solved by Newton - Raphson method and the mass transfer rates are calculated by using Stefan – Maxwell equations. Simulation is written for nonequilibrium and equilibrium model for three-phase distillation to compare each other. Results of experiments are agreed well with the simulation results of both equilibrium and nonequilibrium models. A summary of the numerical methods used in the different studies are given in Table 2.2.

**Table 2.2** Summary of Numerical Methods Used in Different Simulation Studies.

| <b>Author(s)</b>                | <b>Used Method</b>   |
|---------------------------------|--|
| Krishnamurthy and Taylor (1985) | Newton's Method  |
| Wajge et al.(1997)              | Finite difference method and orthogonal collocation method |
| Karacan et al.(1998)            | Orthogonal Collocation on Finite Elements                  |
| Mori et al.(1999)               | Rate Based Model   |
| Attarakih et al.(2001)          | MATLAB Subroutine ode15s<br>Runge-Kutta-Merson Method      |
| Repke et al.(2004)              | Newton - Raphson Method                                    |

## 2.4 Optimal Operation Policy

Batch distillation, which is usually preferred to separate fine chemicals and biochemicals, has the advantage of separating components to product and slop cut tanks by using a single distillation column. The number of product and slop cut tanks increases as the number of distilled component increases. For example, for binary distillation, two product tanks and one slop cut tank are necessary, while in the separation of three components, three product tanks and two slop cut tanks are needed. Also, it is possible to use reboiler tank to collect the heaviest compound as a product tank to decrease investment on tanks.

In the separation of components in different product tanks optimal control criteria can be considered. In these optimal studies one of the following or all can be selected as criteria (Diwekar, 1996):

- maximum distillate,
- minimum time,
- maximum profit.

In the criteria of maximum distillate, the aim is to maximize the amount of distillate for a given time. For the criteria of the minimum time, the time which is necessary to separate a specified amount of feed for a specified concentration is minimized. Lastly, for the criteria of the maximum profit, a function of profit for a concentration of distillate is maximized (Diwekar, 1996).

In batch distillation, there are several parameters that can be optimized like “the size of the initial charge to the still pot and the reflux ratio as a function of time” (Luyben, 1988). By optimizing reflux ratio, the three optimal control criteria explained above can be handled together in a distillation column.

There have been several investigations on the problems specified above to find optimal operation of a batch distillation since 1960s. In almost all studies, an optimized function with its equality and inequality constraints is determined and solved using different solution approaches.

The earliest study on optimal operation is carried by Converse and Gross (1963). It is investigated that the product maximization for a binary batch distillation by using three different techniques; dynamic programming, calculus of variation and Pontriagin’s maximum principle. It is figured out that the use dynamic programming is advantageous and more understandable compared to the other techniques.

The latter study is performed by Coward (1967) who focused on optimization of time for binary batch distillation by using Pontryagin principle. The aim is to find the reflux ratio which gives the maximum Hamiltonian for the parameters of  $p_1$ ,  $p_2$ ,  $B$  and  $x_B$ . The equations are solved with the approach of Simpson's Rule by using subroutines for constant overheads, constant reflux ratio and optimal reflux policy. The results obtained from this principle are compared with those obtained by utilizing the calculus of variation for different examples. It is concluded that the reflux ratio which gives minimum time is between the curves of constant reflux ratio and constant overheads operation.

As a different optimization approach, Luyben (1988) extended his earlier study of capacity factor for product maximization in binary distillation to multicomponent distillation. It is demonstrated that by using capacity factor, not only investments on energy and design, but also parameters like number of trays and reflux ratio can be optimized. Luyben (1988) indicated in his study the applicability of the optimal configuration.

The use of capacity factor approach examined by Bonny (1999) with new operating parameters for a multicomponent batch distillation column by using cyclohexane – toluene – heptane mixtures with or without recycling of slop cuts. The maximization of amount of product is studied by varying reflux ratio policy. Reflux ratio is either constant or is changing in a piecewise constant function throughout the operation time. It is concluded that the use of reflux ratio as a piecewise constant function increased the production rate. Also, it is found that the use of recycling has no significant effect on the production rate.

Increasing demand on the use of the batch distillation in industries in recent years caused researches to focus mostly on maximum profit problem. Kim (1999) investigated for the optimal design and optimal operation of a batch distillation by using dynamic model rather than shortcut model or quasi steady state model. Comparison between dynamic models with others is figured out that design with dynamic model gives higher capacity on the aspect of maximizing product. Also, it is demonstrated that the use of exponential reflux ratio with constant reflux ratio profiles improves the design of the column with respect to the use of constant reflux ratio only. For the optimal operation, maximum profit function is analyzed by using costs of feed, product and steam. 23 are found as an optimum tray number by examining the various data related to maximum profit obtained for different tray numbers. Moreover, recycling of the slop cut products to reboiler to maximize the amount of the product is applied and it is found that this work differ slightly the results obtained for a fresh feed optimization.

Miladi and Mujtaba (2004) searched the optimum design and operation points for a binary batch distillation by using simulated annealing approach (SA). The effect of vapor boil up rate and number of stages investigated to find optimum design, while reflux ratio and batch time are configured for optimum operation by fixing the demand on the amount of product. In the study of Miladi and Mujtaba (2004), optimization of vapor boil up rate is firstly taken into consideration. It is concluded that fixing of vapor boil up rate smaller or bigger than the optimum vapor boil up rate is important to obtain maximum profit in a given production time. Moreover, the studies are carried for unlimited demand of the products. The results demonstrated much more profit can be obtained by using unlimited demand of the products. However, unlimited or unplanned production of product can cause undesirable significant losses.

A different optimal policy has been raised by Low and Sørensen (2005). They found that optimal configuration besides optimum design and operation of a batch distillation is a valuable tool for the estimation of reasonable design and configuration for binary and multicomponent systems by using genetic algorithm and penalty function approaches. There are three choices for optimal configuration: regular, inverted and multi vessel. The aim to find optimum point is to maximize profitability index, which includes "production revenue, capital and operational costs". The model based on discrete – continuous dynamic optimization is examined for cyclohexane – toluene, n-heptane – toluene and pentane, hexane, heptane, octane mixtures. It is concluded that the configuration of distillation column is dependent on the feed composition for binary mixtures. If the feed mixture contains higher fraction of the light component, the use of inverted column is more profitable, and if the feed mixture contains higher fraction of the heavy component, then the use of regular column is more profitable. On the other hand, for the separation of multicomponent mixtures, multivessel configuration is found to be more profitable rather than regular and inverse column configuration.

## **CHAPTER 3**

### **MULTICOMPONENT BATCH PACKED DISTILLATION COLUMN MODELING**

In this chapter, modeling studies of a Multicomponent Batch Packed Distillation Column (MBPDC) with random packing material are presented. Section 3.1 is devoted to model assumptions. While in Section 3.2, model equations of total, component mass and energy balances for packed section, reflux – drum – condenser and reboiler are given together with the used correlations. Also, in this section, calculation of loading and flooding points, liquid holdups, pressure drops and solution proposal for partial differential equations and thermodynamic relationships are discussed in detail. In Section 3.3, information about required data for simulation is explained. Lastly, modeling of storage tanks is given in Section 3.4.

#### **3.1 Model Assumptions**

In the proposed batch packed distillation column model, following assumptions are considered (Alkaya (1990)):

- Negligible vapor holdup throughout the system,
- Equimolar counter diffusion between the phases,
- Adiabatic column operation and negligible temperature change throughout the packed section,
- Dominancy of overall vapor phase mass transfer coefficient to individual liquid and vapor film mass transfer coefficients,
- Negligible effect of maldistribution of liquid and vapor flow rates.



The most important assumption that can make significant changes in dynamic modeling of a packed distillation is negligible vapor holdup. At low pressures, although the vapor volume is large, the number of moles is usually small because the vapor density is much smaller than the liquid density. Choe and Luyben (1987) stated that, at low pressure operations (up to 5atm) this assumption can be made since liquid to vapor density ratio is high. However, at high pressure operations, this ratio is so small that with this assumption the model can be inaccurate.

The existence of equimolar counter diffusion between phases results in constant vapor and liquid rates through the packed section. Also, the assumption of adiabatic operation and negligible temperature change in a differential element leads to omission of energy balance equations for the packed section (Luyben, 1989).

Based on equimolar overflow, the mass flux rate (Equation 3.1) and relationship between mass transfer coefficients (Equation 3.2) can be written as (Geankoplis, 2003);

$$N = k_V(y_V - y_I) = k_L(x_I - x_L) \quad (3.1)$$

$$\frac{1}{K_V} = \frac{1}{k_V} + \frac{m}{k_L} \quad (3.2)$$

Patwardhan and Edgar (1993) stated that, "in distillation applications, the controlling resistance to mass transfer is usually in the vapor film, and the use of the overall vapor phase mass transfer coefficient is more common." Therefore, effect of the liquid phase on the overall coefficient can be considered negligible resulting in,

$$\frac{1}{K_V} = \frac{1}{k_V} \quad (3.3)$$

Maldistribution of vapor and liquid flows throughout the packed distillation column is known as an effect which reduces the performance of the column (Higler et al., 1999). The effect of liquid flow maldistribution is generally observed at the top of the packed section. Liquid flow on the wall can be prevented by using side wipers or redistributors and arranging the ratio of the diameters of the column and packing material (Perry et al., 1997). Thus, the effect of maldistribution of flow rates can be neglected.

Additional assumptions basing on the previous studies (Alkaya (1990) and Yıldız (2002)) are as follows:

- Constant liquid holdup in the reflux drum,
- Change of composition only in z direction, and no radial variation,

- Perfect mixing in the reflux drum, reboiler and differential element of packed section,
- Total condenser.

### **3.2 Batch Packed Distillation Column Dynamic Modeling**

Modeling of batch distillation systems is complex due to unsteady state nature of the system. Consequently, dynamics is inherently included in any modeling issue for the batch system.

Appropriate conservation equations, derived equations for the determination of flow rates at finite and total reflux operations, correlations for the prediction of dry, irrigated and flooding pressure drops, molar hold up and physical properties, equations for estimation of vapor liquid equilibrium compositions are given below.

#### **3.2.1 Conservation Equations**

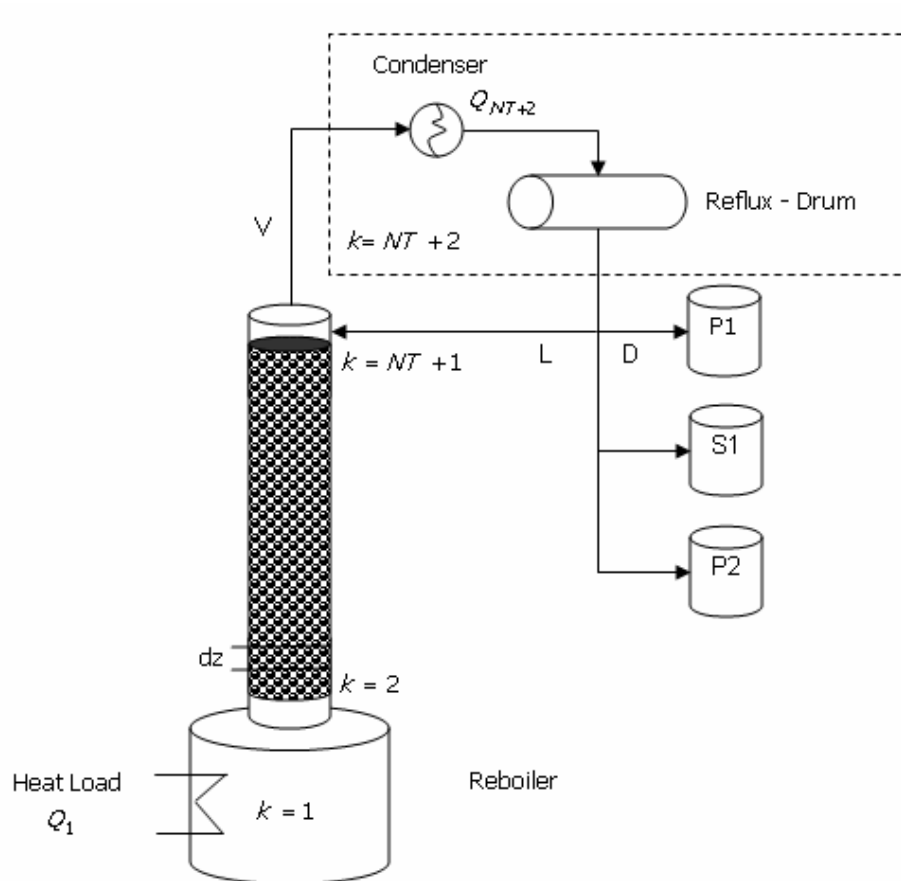
The modeling of a batch packed distillation column is very similar to that of batch plate distillation column. These equations are nearly the same except with those about hydraulics. The important features of the model equations for a batch packed and plate column are given in Table 3.1 (Salimi and Depeyre, (1998), Fieg et al. (1994)).

A schematic view of the studied batch packed distillation column is shown in Figure 3.1. In the figure the packed column with its reboiler, condenser, reflux drum and tanks are shown. The parameter  $k$  in Figure 3.1 defines the differential element starting with  $k = 1$  at reboiler.  $P_1$  and  $P_2$  are the product tanks and  $S_1$  shows the slop cut tank. Thus, packed section is considered to be composed of  $NT$  differential elements whose value can be changed arbitrarily and will have an effect on modeling accuracy.

In the present study, discrete element concept is used instead of HETP concept to accomplish dynamic modeling of the batch packed distillation column. In discrete element method, the packed section is divided by  $NT$  into segments of  $\Delta z$  height ( $\Delta z = (\text{Height of packed section}) / NT$ ). For each  $\Delta z$  element energy and mass transfer equations are considered. However, in packed distillation column system transfer operations are continuous. Therefore, the choice of  $\Delta z$  is very important and affects the accuracy of the results.

**Table 3.1** Important Features of Batch Distillation Column Models.

| Equation                | Packed                            | Plate                             |
|-------------------------|-----------------------------------|-----------------------------------|
| Thermodynamics          | 1 Equilibrium                     | 1 Equilibrium                     |
|                         | 2 Non - Equilibrium               | 2 Non - Equilibrium               |
| Mass and Energy Balance | 1 Differential                    | 1 Differential                    |
|                         | 2 Finite Difference Approximation | 2 Finite Difference Approximation |
|                         |                                   |                                   |
| Hydraulics              | 1 Continuous Contact              | 1 Step By Step Contact            |
|                         | 2 Small Liquid Holdup             | 2 Big Liquid Holdup               |
|                         | 3 Small Total Pressure Drop       | 3 Big Total Pressure Drop         |



**Figure 3.1** Schematic View of Studied Batch Packed Distillation Column.

The unsteady state total mass and material balances for a packed differential element shown in Figure 3.2 (Jahromi et al., 1982) can be written as follows for the liquid and vapor phases:

Liquid phase:  $j = 1 \dots NC$

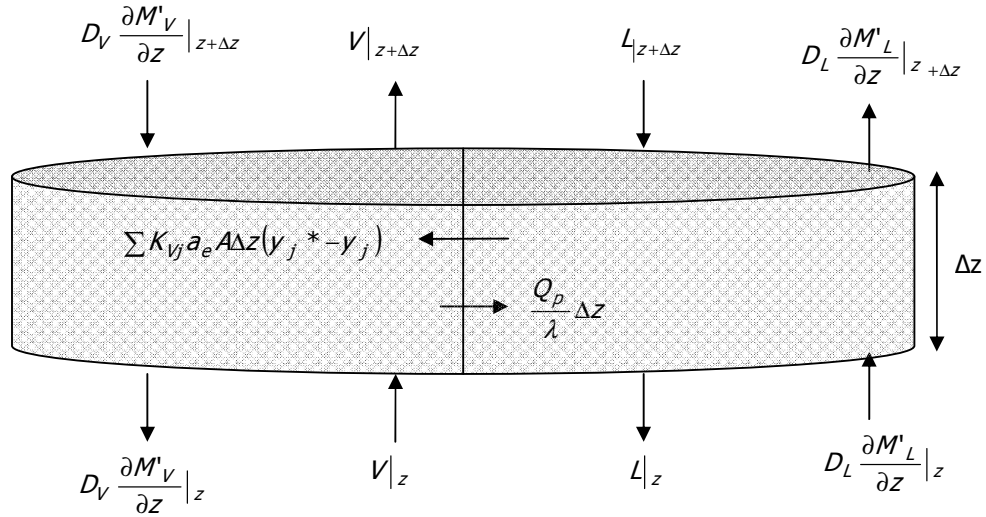
$$\frac{\partial(M'_L)}{\partial t} = \frac{\partial L}{\partial z} + \frac{\partial}{\partial z} \left[ D_L \frac{\partial M'_L}{\partial z} \right] - \sum_j^{NC} K_{Vj} A a_e (y_j^* - y_j) + \frac{Q_P}{\lambda} \quad (3.4)$$

$$\frac{\partial(M'_L x_j)}{\partial t} = \frac{\partial(Lx_j)}{\partial z} + \frac{\partial}{\partial z} \left[ D_L \frac{\partial(M'_L x_j)}{\partial z} \right] - K_{Vj} A a_e (y_j^* - y_j) + \frac{Q_P}{\lambda} x_j^* \quad (3.5)$$

Vapor phase:  $j = 1 \dots NC$

$$\frac{\partial(M'_V)}{\partial t} = -\frac{\partial V}{\partial z} + \frac{\partial}{\partial z} \left[ D_V \frac{\partial(M'_V)}{\partial z} \right] + \sum_j^{NC} K_{Vj} A a_e (y_j^* - y_j) - \frac{Q_P}{\lambda} \quad (3.6)$$

$$\frac{\partial(M'_V y_j)}{\partial t} = -\frac{\partial(Vy_j)}{\partial z} + \frac{\partial}{\partial z} \left[ D_V \frac{\partial(M'_V y_j)}{\partial z} \right] + K_{Vj} A a_e (y_j^* - y_j) - \frac{Q_P}{\lambda} x_j^* \quad (3.7)$$



**Figure 3.2** Schematic View of a Differential Section of the Column for Material Transfer.

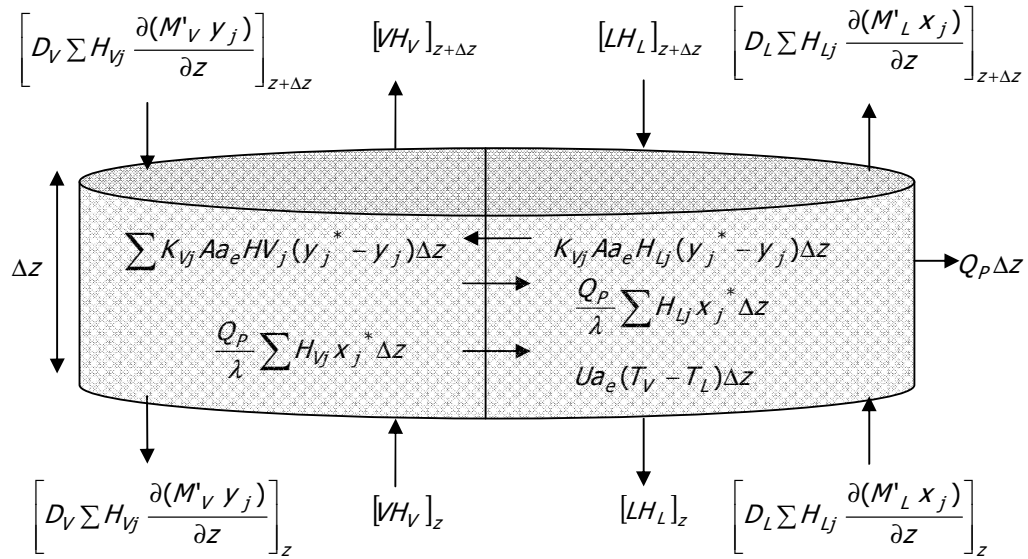
The unsteady state energy balances of a packed differential element which is demonstrated in Figure 3.3 (Jahromi, et al., 1982) are as follows;

Liquid phase:  $j = 1 \dots NC$

$$\begin{aligned} \frac{\partial(M'_L h)}{\partial t} = & \frac{\partial(Lh)}{\partial z} + \frac{\partial}{\partial z} \left[ D_L \sum_j h_j \frac{\partial(M'_L x_j)}{\partial z} \right] - \sum_j^{NC} K_{Vj} A a_e h_j (y_j^* - y_j) \\ & + \frac{Q_P}{\lambda} \sum_j h_j x_j^* + U a_e (T_V - T_L) \end{aligned} \quad (3.8)$$

Vapor phase:  $j = 1 \dots NC$

$$\begin{aligned} \frac{\partial(M'_V H)}{\partial t} = & -\frac{\partial(VH)}{\partial z} + \frac{\partial}{\partial z} \left[ D_V \sum_j H_j \frac{\partial(M'_V y_j)}{\partial z} \right] + \sum_j H_j K_{Vj} A a_e (y_j^* - y_j) \\ & - \frac{Q_P}{\lambda} \sum_j H_j x_j^* - U a_e (T_V - T_L) \end{aligned} \quad (3.9)$$



**Figure 3.3** Schematic View of a Differential Section of the Column for Energy Transfer.

However, considering equimolar counter diffusion model, adiabatic process and constant temperature profile energy balance equations can be omitted and the mass transfer equations (Equations 3.4 - 3.7) are simplified as:

Liquid phase:

$$M'_L \frac{\partial x_j}{\partial t} = L \frac{\partial x_j}{\partial z} - K_{Vj} A a_e (y_j^* - y_j) \quad (3.10)$$

Vapor phase:

$$0 = -V \frac{\partial y_j}{\partial z} + K_{Vj} A a_e (y_j^* - y_j) \quad (3.11)$$

To evaluate  $K_V$  and  $a_e$ , Onda correlation (Wang et al., 2005) for volumetric mass transfer and effective interfacial area are used and are given as followings, respectively,

$$k_V = c \left( \frac{D_V}{a_p d_p^2} \right) \text{Re}_V^{0.7} \text{Sc}_V^{1/3} \quad (3.12)$$

$$\frac{a_e}{a_p} = 1 - \exp \left[ -1.45 \left( \frac{\sigma_c}{\sigma_L} \right)^{0.75} \text{Re}_L^{0.1} \text{Fr}_L^{-0.05} \text{We}_L^{0.2} \right] \quad (3.13)$$

where,

$$\text{Re}_L = \frac{LMW}{3600 A a_p \mu_L} \quad (3.14)$$

$$\text{Re}_V = \frac{VMW}{3600 A a_p \mu_V} \quad (3.15)$$

$$\text{Sc}_V = \frac{\mu_V}{\rho_V^* D_V} \quad (3.16)$$

$$\text{Fr}_L = \frac{L^2 a_p MW^2}{(3600)^2 \rho_L^2 A^2 g} \quad (3.17)$$

$$\text{We}_L = \frac{L^2 MW^2}{(1321.6) a_p \rho_L \sigma A^2 g} \quad (3.18)$$

The unit conversion of mass transfer coefficient from m/s to mol/ (h.m<sup>2</sup>) is done as:

$$K_V = 3600 \frac{P}{R_g T} k_V \quad (3.19)$$

Initial and boundary conditions of Equations 3.10 and 3.11 are:

$$x(z,0) = x(z) \quad (3.20)$$

$$y(z,0) = y(z) \quad (3.21)$$

$$y(0,t) = y_B \quad (3.22)$$

The approach used by Attarakih et al. (2001) is applied to convert partial differential equations (PDEs) to ordinary differential equations (ODEs) as follows:

For  $i = 1 \dots DL$  and for  $j = 1 \dots NC$

$$\frac{\partial x}{\partial z} = \frac{x_{i+1} - x_i}{\Delta z} \quad (3.23)$$

$$\frac{\partial y}{\partial z} = \frac{y_{i+1} - y_i}{\Delta z} \quad (3.24)$$

$$y^* = \frac{y^*(x_i) + y^*(x_{i+1})}{2} \quad (3.25)$$

$$y = \frac{y_{i-1} + y_i}{2} \quad (3.26)$$

By substituting Equations 3.23, 3.24, 3.25 and 3.26 to Equations 3.10 and 3.11, one can obtain,

$$M'_L \frac{dx_{i,j}}{dt} = L \frac{x_{i+1,j} - x_{i,j}}{\Delta z} + K_{Vj} Aa_e \left( \frac{y_{i-1,j} + y_{i,j}}{2} - \frac{y^*(x_{i,j}) + y^*(x_{i+1,j})}{2} \right) \quad (3.27)$$

$$0 = -V \frac{y_{i+1,j} - y_{i,j}}{\Delta z} - K_{Vj} Aa_e \left( \frac{y_{i-1,j} + y_{i,j}}{2} - \frac{y^*(x_{i,j}) + y^*(x_{i+1,j})}{2} \right) \quad (3.28)$$

Then Equations 3.27 and 3.28 are arranged into state equations as;

$$\frac{dx_{i,j}}{dt} = \frac{L}{M'_L} \frac{x_{i+1,j} - x_{i,j}}{\Delta z} + \frac{K_{Vj} Aa_e}{2M'_L} \left( (y_{i-1,j} + y_{i,j}) - (y^*(x_{i,j}) + y^*(x_{i+1,j})) \right) \quad (3.29)$$

$$y_{i+1,j} = -\frac{K_{Vj} Aa_e \Delta z}{V} \left( \frac{y_{i-1,j} + y_{i,j}}{2} - \frac{y^*(x_{i,j}) + y^*(x_{i+1,j})}{2} \right) + y_{i,j} \quad (3.30)$$

Since it is assumed that there is only liquid molar holdup in the column, the subscripts of the liquid molar holdup may be neglected. Liquid molar holdup can be written as,

$$M_L = M \quad (3.31)$$

Total mass, material and energy balances for the other parts of the distillation column can be given as;

Reflux – drum – condenser system ( $k = NT+2$ ):  $j = 1 \dots NC$

$$\frac{dM_{NT+2}}{dt} = V - L - D \quad (3.32)$$

$$\frac{d(M_{NT+2} x_{NT+2,j})}{dt} = Vy_{NT+2,j} - Lx_{NT+2,j} - Dx_{NT+2,j} \quad (3.33)$$

$$\frac{d(M_{NT+2}h_{NT+2})}{dt} = VH_{NT+1} - Lh_{NT+2} - Dh_{NT+2} - Q_{NT+2} \quad (3.34)$$

where D is the distillate flow rate and  $Q_{NT+2}$  is the heat removed from the condenser.

Reboiler (k = 1):

$$\frac{dM_1}{dt} = L - V \quad (3.35)$$

$$\frac{d(M_1x_{1,j})}{dt} = Lx_{2,j} - Vy_{1,j} \quad (3.36)$$

$$\frac{d(M_1h_1)}{dt} = Lh_2 - VH_1 + Q_1 \quad (3.37)$$

where  $Q_1$  is the heat given to the reboiler.

Using the procedure given by Yıldız (2002) Equations 3.35 and 3.36 transformed into a state equation for compositions in the reboiler as:

$$\frac{dx_{1,j}}{dt} = [L(x_{2,j} - x_{1,j}) - V(y_{1,j} - x_{1,j})] / M_1 \quad (3.38)$$

Thus using the constant molar liquid holdup assumption in the reflux drum, Equation 3.32 simplifies as,

$$V = L + D \quad (3.39)$$

Using the procedure given by Yıldız (2002) Equations 3.33 and 3.39 are transformed into a state equation for compositions at the distillate compositions.

$$\frac{dx_{NT+2,j}}{dt} = [V(y_{NT+2,j} - x_{NT+2,j})] / M_{NT+2} \quad (3.40)$$

### 3.2.2 Determination of Flow Rates at Finite External Reflux Ratio

The knowledge of the vapor and liquid flow rates are necessary to calculate mole fractions of liquid and vapor phases at any location of the packed distillation column. The estimation of the flow rates at finite external reflux ratio will be explained below by using the procedure of Yıldız (2002) in rigorous modeling with the assumption of equimolar counter diffusion.

Vapor rate, V, can be obtained by using the external reflux ratio definition ( $R = L/D$ ) and Equation 3.39 as;

$$V = L + D = D(R + 1) \quad (3.41)$$



Thus, in reflux – drum – condenser, substituting Equation 3.41 into Equation 3.34 with the assumption of constant hold-up and total condenser the following can be obtained;

$$M_{NT+2} \frac{dh_{NT+2}}{dt} = VH_{NT+1} - Lh_{NT+2} - Dh_{NT+2} - Q_{NT+2} \quad (3.42)$$

and inserting Equation 3.39 into Equation 3.42;

$$M_{NT+2} \frac{dh_{NT+2}}{dt} = V(H_{NT+1} - h_{NT+2}) - Q_{NT+2} \quad (3.43)$$

Equation 3.43 can be written as;

$$M_{NT+2} \delta_t(h_{NT+2}) = V(H_{NT+1} - h_{NT+2}) - Q_{NT+2} \quad (3.44)$$

and considering,

$$\delta_t(*) = \frac{*(t_k) - *(t_{k-1})}{t_k - t_{k-1}} \quad (3.45)$$

where \* is a dummy function (Yıldız (2002)).

Further substituting Equation 3.41 into Equation 3.44, heat removed from the condenser can be written as;

$$Q_{NT+2} = D(R + 1)(H_{NT+1} - h_{NT+2}) - M_{NT+2} \delta_t(h_{NT+2}) \quad (3.46)$$

Total energy balance over the packed distillation column can be written as;

$$Q_1 - Q_{NT+2} - Dh_{NT+2} = \sum_{n=1}^{NT+2} \frac{d(M_n h_n)}{dt} \quad (3.47)$$

Thus, distillate rate can be obtained by substituting Equation 3.46 into Equation 3.47;

$$D = \frac{Q_1 - \sum_{n=1}^{NT+2} \frac{d(M_n h_n)}{dt}}{(R + 1)H_{NT+1} - Rh_{NT+2}} \quad (3.48)$$

However, to find the distillate rate, D, one must know the liquid molar holdup for each section,  $M_n$ , which will be explained in Section 3.2.5.

### 3.2.3 Determination of Flow Rates at Total Reflux

In the previous section, the estimation of flow rates for finite external reflux ratio is described. However, to find flow rates at total reflux ( $R = \infty$ ), further analysis must be done, since  $D = 0$  at total reflux and  $V = L$  from Equation 3.41.

If the energy balance is made on the packed distillation column without reflux – drum – condenser, following equation is obtained:

$$Q_1 + Lh_{NT+2} - VH_{NT+1} = \frac{d(M_{NT+2}h_{NT+2})}{dt} + \frac{d(M_1h_1)}{dt} \quad (3.49)$$

Substituting  $V = L$  into Equation (3.49) gives,

$$L = \frac{Q_1 - \frac{d(M_1h_1)}{dt} - \frac{d(M_{NT+2}h_{NT+2})}{dt}}{(H_{NT+1} - h_{NT+2})} \quad (3.50)$$

By using Equation 3.50, flow rates at total reflux can be calculated. As stated before, to solve Equation 3.48 and 3.50, molar holdups at reboiler, condenser and packed section should be known.

### 3.2.4 Determination of Loading and Flooding Points

Operation of packed distillation column can be analyzed in three regions: preloading, loading and flooding. Loading and flooding points are two important operational points for the simulations, because pressure drop and liquid holdup throughout the column depends directly on these points. Flooding point is the upper limit of the packed distillation column operation, while loading point is a measure that demonstrates generation of high pressure drop. Billet and Schultes (1999) demonstrated that loading point is considered to occur approximately 65 – 70% of the flooding point. In this study, the occurrence of loading point and design gas rate are chosen as at 60% and 40% of flooding point, respectively. Prediction of flooding point will be explained in Section 3.2.5.

### 3.2.5 Calculation of Molar Hold up and Pressure Drop

Several investigations are made on the evaluation of liquid molar hold ups. Some of them include the effect of surface tension, viscosity, vapor and liquid flow rates. In this study, the correlation derived by Stichlmair et al. (1989) is used. Stichlmair et al. (1989) stated that several liquids for liquid molar holdup up to loading point show the same behavior, which is a function of Froude number. Equation for liquid holdup for a given packed section for preloading region is as follows;

$$M'_L = (A\rho_L h_o) / MW \quad (3.51)$$

where,

$$h_o = 0.555Fr_L^{1/3} \quad (3.52)$$

$$Fr_L = u_L^2 \frac{a_p}{g\varepsilon^{4.65}} \quad (3.53)$$

As Stichlmair et al. (1989) stated, above the loading point holdup increases with an increase in gas rate at constant liquid rate. The liquid is restrained by the friction forces and static pressure gradient and this effect is demonstrated for liquid holdup at loading point as follows;

$$h_L = h_o \left( 1 + 20(\Delta P_{irr} / (Z\rho_L g))^2 \right) \quad (3.54)$$

In this study, design gas velocity is selected as the 40% of the flooding velocity so the relationship for molar holdup below loading point,  $h_o$ , is used.

In packed distillation column, there are three types of pressure drops: dry, irrigated and flooding. Correlations derived by Stichlmair et al. (1989) are used in this study and given as follows;

$$\Delta P_{dry} / z = 3 / 4 f_o \left[ (1 - \varepsilon) / \varepsilon^{4.65} \right] \rho_g u_g^2 / d_p \quad (3.55)$$

$$\Delta P_{irr} / \Delta P_{dry} = \frac{\left\{ 1 - \varepsilon \left[ 1 - h_o / \varepsilon (1 + 20[\Delta P_{irr} / (Z\rho_L g)]^2) \right] (1 - \varepsilon)^{-1} \right\}^{(2+c)/3}}{\left\{ 1 - h_o / \varepsilon (1 + 20[\Delta P_{irr} / (Z\rho_L g)]^2) \right\}^{4.65}} \quad (3.56)$$

where,

$$f_o = C_1 / Re_g + C_2 / Re_g^{1/2} + C_3 \quad (3.57)$$

$$c = \frac{-C_1 / Re_g - C_2 / (2Re_g^{1/2})}{f_o} \quad (3.58)$$

$C_1$ ,  $C_2$  and  $C_3$  are constants related to the packing material and the constants for dumped ceramic packing materials are given in Table B.1 (Appendix B). At flooding point, the pressure is calculated from the following equation,

$$\left( \frac{\Delta P_{irr}}{Z\rho_L g} \right)_f^{-2} - \frac{40 \frac{2+c}{3} h_o}{1 - \varepsilon + h_o \left( 1 + 20[\Delta P_{irr} / (Z\rho_L g)]_f^2 \right)} - \frac{186 h_o}{\varepsilon - h_o \left( 1 + 20[\Delta P_{irr} / (Z\rho_L g)]_f^2 \right)} = 0 \quad (3.59)$$

Calculation of flooding pressure drop is an iterative process. First, a gas rate is assumed and then, dry pressure drop is calculated and, for the same gas rate at a fixed liquid rate, irrigated pressure drop is calculated. Then, Equation 3.59 is checked. If Equation 3.59 is not

satisfied, another gas rate is assumed and the same calculations are done until the Equation 3.59 is satisfied.

Since it is assumed that design gas rate below the loading point and that the pressure drop profile is linear through packed section, equation for pressure drop profile can be written as;

$$P(z) = P_B + z * \frac{\Delta P_{irr}}{Z} \quad (3.60)$$

### 3.2.6 Predictions of Physical Properties

Parameters like viscosity, surface tension, diffusivity coefficient are necessary to estimate mass transfer coefficient and effective interfacial area. These parameters generally are functions of temperature, the number of components in the mixture and the type of the components. Thus, in this study, for hydrocarbons, prediction of these parameters are done at low pressures by most commonly used correlations, and for polar mixtures, these parameters are kept constant because, correlations of physical properties for polar mixtures are complex and the usage of these correlations in the simulation code will not be user friendly. Critical surface tension of materials is given by Perry et al. (1984). A list of the correlations to predict physical properties for hydrocarbons at low pressures is tabulated in Appendix D with their percent errors (Perry et al., 1997).

### 3.2.7 Algebraic Equations and Thermodynamics

Starting point of the all the predictions of vapor – liquid equilibrium compositions is as follows, for  $j=1\dots NC$ ,

$$\hat{f}_j^V = \hat{f}_j^L \quad (3.61)$$

In this study, there are two types of mixtures to be analyzed: hydrocarbons and polar mixtures. For hydrocarbons at low pressures, to predict vapor liquid equilibrium (VLE) relationships Peng Robinson Equation of State (PR-EOS) is utilized for both, vapor and liquid phases. The use of cubic equations of state for vapor and liquid phases is known as Phi- Phi ( $\Phi - \Phi$ ) approach and using Equation 3.61 equilibrium relationship can be written as, for  $j=1\dots NC$ ,

$$y_j \hat{\phi}_j^V = x_j \hat{\phi}_j^L \quad (3.62)$$

Detailed information on calculation of VLE by using PR-EOS can be found in the studies of Alkaya (1990), Dokucu (2002) and Yıldız (2002).

On the other hand, PR-EOS must not be used to predict vapor liquid equilibrium for liquid phase of polar mixtures (Sandler, 1999). Hence, for liquid phase, an activity coefficient model and for the vapor phase a cubic equation of state will be used and this is known as the Gama – Phi ( $\gamma$ - $\phi$ ) approach. Then, Equation 3.61 can be written as,

$$y_j \hat{\phi}_j P = x_j \gamma_j f_j \quad (3.63)$$

Bahar (2007) showed that, the use of Non Random-Two Liquid (NRTL) activity coefficient model gives best results for the prediction of compositions of polar mixtures. Therefore, in this study, for polar mixtures NRTL activity coefficient will be used to predict liquid phase and, PR – EOS will be used to predict vapor phase compositions at equilibrium. NRTL model equations for the excess Gibbs energy for condensed phase binary systems developed by Renon and Prausnitz (1968) are given in Equations 3.64 - 3.69 (Tester and Modell, 1997).

For  $i = 1 \dots NC, j = 1 \dots NC$

$$\ln \gamma_1 = x_2^2 \left[ \tau_{21} \left( \frac{G_{21}}{x_1 + x_2 G_{21}} \right)^2 + \frac{\tau_{12} G_{12}}{(x_2 + x_1 G_{12})^2} \right] \quad (3.64)$$

$$\ln \gamma_2 = x_1^2 \left[ \tau_{12} \left( \frac{G_{12}}{x_2 + x_1 G_{12}} \right)^2 + \frac{\tau_{21} G_{21}}{(x_1 + x_2 G_{21})^2} \right] \quad (3.65)$$

where,

$$\tau_{ji} = \frac{g_{ji} - g_{ii}}{RT} \quad (3.66)$$

$$\tau_{ii} = \tau_{jj} = 0 \quad (3.67)$$

$$G_{ji} = \exp(-\alpha_{ji} \tau_{ji}) \quad (3.68)$$

$$G_{ii} = G_{jj} = 1.0 \quad (3.69)$$

Extension of NRTL model to multicomponent systems is given in Equation 3.70 (Tester, et. al., 1997).

$$\ln \gamma_i = \frac{\sum_{j=1}^n \tau_{ji} G_{ji} X_j}{\sum_{k=1}^n G_{ki} X_k} + \sum_{j=1}^n \left[ \frac{G_{ij} X_j}{\sum_{k=1}^n G_{kj} X_k} \right] \tau_{ij} - \frac{\sum_{m=1}^n \tau_{mj} G_{mj} X_m}{\sum_{k=1}^n G_{kj} X_k} \quad (3.70)$$

NRTL model parameters used in this study for polar mixtures are given at Table D.5 (Maier, et al., 1998).

The sum of compositions in terms of mole fractions is equal to 1.0 in liquid and vapor phases, as follows,

$$\sum_{j=1}^{NC} x_j = 1 \quad (3.71)$$

$$\sum_{j=1}^{NC} y_j = 1 \quad (3.72)$$

Binary interaction parameters that are used in the simulation code, which "has been introduced to obtain better agreement in mixture equation of state calculations" (Sandler, 1999), used for this study are given in Table 3.2 (Bahar (2007), Yıldız (2002)).

**Table 3.2** Binary Interaction Parameters,  $b_{ij}$ .

| $b_{ij}$    | Ethanol | Water  | Cyclohexane | N - Heptane | Toluene |
|-------------|---------|--------|-------------|-------------|---------|
| Ethanol     | 0.0     | -0.935 | -           | -           | -       |
| Water       | -0.935  | 0.0    | -           | -           | -       |
| Cyclohexane | -       | -      | 0.0         | 0.0         | 0.0     |
| N - Heptane | -       | -      | 0.0         | 0.0         | 0.0     |
| Toluene     | -       | -      | 0.0         | 0.0         | 0.0     |

### 3.3 Initial Conditions for the Simulation

The equations given above are formulations of the mathematical model of a multicomponent batch packed distillation column. In order to be able to solve these equations, MATLAB and FORTRAN programming languages are used.

Initial conditions are necessary to solve ordinary differential equations. In this study, the feed is charged to the column from the top at its boiling point. Therefore, at the beginning of the operation, whole column with its reboiler and reflux – drum – condenser part is assumed to have the same feed compositions (Yıldız (2002)), that is,

$$x_{ij} = x_j^{feed} \text{ for } i = 1 \dots NT + 2$$

$$j = 1 \dots NC \quad (3.73)$$

Also, the initial flow rates are estimated by using the following equations (Yıldız (2002)),

$$V_i = \frac{Q_1}{(H_1 - h_2)} \text{ for } i = 2 \dots NT + 2 \quad (3.74)$$

$$L_i = V_{NT+2} \text{ for } i = 1 \dots NT + 1 \quad (3.75)$$

### 3.4 Storage Tank Models

In batch distillation, distilled products can be collected at separate tanks. These tanks are product cut tanks and slop cut tanks. Product cut tanks are the tanks where products are collected at specified purity levels or amounts, while slop cut tanks are the tanks where waste material of the distillate is collected i.e. slop cut tanks are utilized to collect the distilled mixture whose purity is not at specified purity of any products.

In order to start the operation, the fresh feed is introduced to the system and then necessary heat load is given to the reboiler. After the contents of the reboiler started to boil, vapor flows through the packed section and moves through the condenser. If the system operates for a reflux ratio other than total reflux, then, some distilled material is collected at the product tank 1 until the composition of contents of the product tank 1 reaches a specified purity level. After specified purity level of the lightest compound in product tank 1 drops, distilled material is collected at the slop cut tank 1 until the desired purity level of the second compound is reached. Then, second product will be started to collect at product tank 2. This operation continues for product tanks equal to the number of components. Reboiler can be used as product tank to collect the heaviest compound to decrease investment on product and slop cut tanks.

Modeling of the storage tanks, which is same with the study of the Yıldız (2002), consists of calculating holdups and compositions of product and slop cut tanks. Holdup of a tank at any time is calculated by integrating distillate flow rate from the time of the start to the end of accumulating material in the tank. For the estimation of compositions in a tank, firstly, the multiplication of liquid composition of the distilled material with distillate flow rate is integrated from the time of the start to the end of accumulating material in the tank. Then, this integration is divided to the holdup in the tank.

### 3.5 Model Equations

All the state and algebraic equations are listed in Table 3.3 in order, starting from reboiler to reflux-drum-condenser.

**Table 3.3** Summary of MPDC Model Equations.

| <b>State Equations</b>   |  |
|--|--|
| <b><u>Reboiler</u></b>   |  |
| $\frac{dx_{1,j}}{dt} = [L(x_{2,j} - x_{1,j}) - V(y_{1,j} - x_{1,j})] / M_1$  |  |
| <b><u>Packed Section</u></b>   |  |
| $\frac{dx_{i,j}}{dt} = \frac{L}{M'_L} \frac{x_{i+1,j} - x_{i,j}}{\Delta z} + \frac{K_{Vj} A a_e}{2M'_L} ((y_{i-1,j} + y_{i,j}) - (y^*(x_{i,j}) + y^*(x_{i+1,j})))$ |  |
| <b><u>Reflux – Drum – Condenser</u></b>  |  |
| $\frac{dx_{NT+2,j}}{dt} = [V(y_{NT+2,j} - x_{NT+2,j})] / M_{NT+2}$   |  |
| <b>Algebraic Equations</b>   |  |
| <b><u>Composition Sums</u></b>   |  |
| $\sum_{j=1}^{NC} x_j = 1$  |  |
| $\sum_{j=1}^{NC} y_j = 1$  |  |
| <b><u>Mass Transfer Correlation</u></b>  |  |
| $K_V = 3600 \frac{P}{R_g T} c \left( \frac{D_V}{a_p d_p^2} \right) (\text{Re}_V)^{0.7} \text{Sc}_V^{1/3}$  |  |
| $\frac{a_e}{a_p} = 1 - \exp \left[ -1.45 \left( \frac{\sigma_c}{\sigma_L} \right)^{0.75} \text{Re}_L^{0.1} \text{Fr}_L^{-0.05} \text{We}_L^{0.2} \right]$          |  |



**Table 3.3** Summary of MPDC Model Equations (Cont'd).

| <b>Algebraic Equations</b>  |
|---|
| <p><b><u>Flowrates at Finite External Reflux Ratio</u></b></p> $R = \frac{L}{D}$ $V = L + D = D(R + 1)$ $D = \frac{Q_1 - \sum_{n=1}^{NT+2} \frac{d(M_n h_n)}{dt}}{(R + 1)H_{NT+1} - Rh_{NT+2}}$         |
| <p><b><u>Flow rates at Total Reflux</u></b></p> $D = 0$ $V = L$ $L = \frac{Q_1 - \frac{d(M_1 h_1)}{dt} - \frac{d(M_{NT+2} h_{NT+2})}{dt}}{(H_{NT+1} - h_{NT+2})}$                                       |
| <p><b><u>Vapor Composition</u></b></p> $y_{i+1,j} = -\frac{K_{Vj} A a_e \Delta Z}{V} \left( \frac{y_{i-1,j} + y_{i,j}}{2} - \frac{y^*(x_{i,j}) + y^*(x_{i+1,j})}{2} \right) + y_{i,j}$                  |
| <p><b><u>Holdup below Loading Point</u></b></p> $h_o = 0.555 F r_L^{1/3}$   |
| <p><b><u>Holdup above Loading Point</u></b></p> $h_L = h_o \left[ 1 + 20(\Delta p_{irr} / (z p_L g))^2 \right]$   |
| <p><b><u>Pressure Drop Profile</u></b></p> $P(z) = P_B + \Delta Z * \frac{\Delta P_{irr}}{Z}$   |
| <p><b><u>Thermodynamic Models &amp; Physical Properties</u></b></p> <p style="text-align: center;">Peng Robinson EOS<br/> NRTL Activity Coefficient Model Equations<br/> Correlations at Appendix D</p> |

## CHAPTER 4

### OPTIMAL OPERATION OF A MULTICOMPONENT BATCH PACKED DISTILLATION COLUMN

In this chapter, optimal operation for maximizing product amount in a multicomponent batch packed distillation column is explained with an approach to find optimum reflux ratio. Problem identification and optimal control formulation are given in Sections 4.3 and 4.4, respectively.

#### 4.1 Capacity Factor

The approach to optimize product amount in a multicomponent batch packed distillation column by using capacity factor (CAP) was developed by William L. Luyben in 1971. "The capacity factor of the batch still is defined as the total specification products produced ( $P_i$ ) divided by the total time of batch" (Luyben, 1988). The formula is given as;

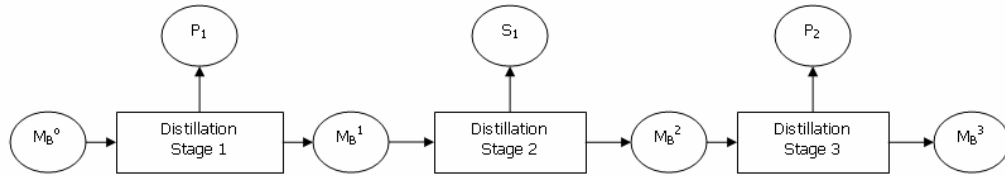
$$CAP = \frac{\sum_{i=1}^{NC} M_{P_i}}{t_T + 0.5} \quad (4.1)$$

where  $t_T$  is the total operation time including total and finite reflux operations. The value of 0.5 hour in the denominator is given for emptying and charging the feed to the still pot since in the study, the distillation column is considered to be operated for 24 h for all days of a year. The total feed fed to still pot initially is equal to the sum of the amount of product and slop cut tanks as given by Luyben (1988) (Equation 4.2).

$$M_B = \sum_{i=1}^{NC} M_{P_i} + \sum_{i=1}^{NC-1} M_{S_i} \quad (4.2)$$

## 4.2 Problem Statement

The system operation can be figured out by using "a state task network (STN) where a state (denoted by a circle) represents a specified material, and a task (rectangular box) represents the operational step which transforms the input state(s) into the output state(s)" (Mujtaba, and Macchietto, 1993). STN of the optimization problem is given in Figure 4.1.



**Figure 4.1** STN of the Optimization Problem.

In Figure 4.1, fresh feed,  $M_B^0$ , is fed to reboiler at the start of the system operation and it is desired to have maximum distilled amount for product 1, the lightest compound, at specified desired purity level,  $x_1^{**}$ . The remaining molar holdup of the reboiler,  $M_B^1$ , is going to be processed until the composition of the second compound purity level in the reflux – drum – condenser part reaches to the desired level. Distilled material which does not have any specified purity level of the compounds is collected in the slop cut tank 1,  $S_1$ , during step 2. The remaining molar holdup in the reboiler,  $M_B^2$ , will be distilled until the product purity of the 2<sup>nd</sup> compound in the product tank 2 drops to specified purity level. This process will be going on to the step NC for a given feed charge with NC specifications.

## 4.3 Optimization Problem Formulation

Optimization of reflux ratio profile to maximize distilled product amount in a multicomponent batch packed distillation column is performed by maximizing CAP. Formulation of the problem necessitates the inputs as; column, feed mixture and product specifications, heat load and initial values (reflux ratio, pressure in the reboiler, dummy variables etc.) by using the following representation (Bahar, 2007):

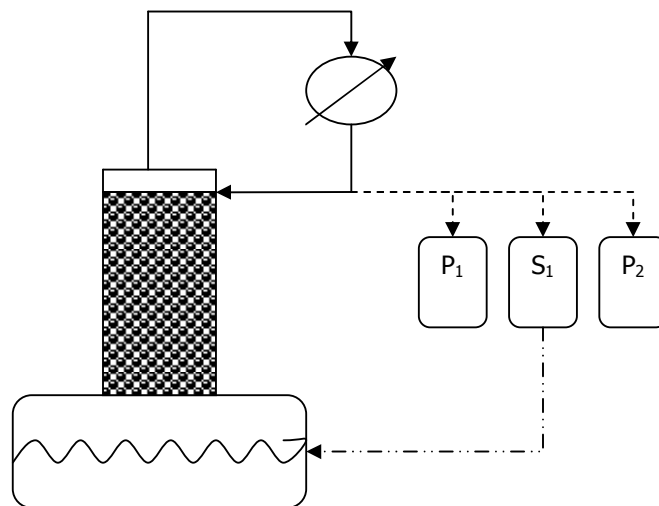
$$\begin{aligned} & \max f(x) \text{ subject to} \\ & 0 \leq R_p \leq 1 \end{aligned} \tag{4.3}$$

where  $R_p$  is the internal reflux ratio ( $L / V$ ) and

$$CAP = f(x) = \frac{\sum_{i=1}^{NC} M_{P_i}}{t_T + 0.5} \quad (4.4)$$

Recycling of the materials in the slop cut tanks for the next batch may be done as shown in the Figure 4.2 to increase product amount. As Luyben (1988) and Bonny (1999) stated recycling of holdups of the slop cut tanks do not have any significant effect on CAP. However, it increased the product amount by 16% for multicomponent batch plate distillation columns.

In this study, also the recycling of molar holdup of slop cut tanks for the next batch is examined to analyze the effect of recycle in packed distillation columns.



**Figure 4.2** Schematic View of Recycling of the Materials in Slop Cut Tank.

## CHAPTER 5

### SIMULATION CODE

In this chapter the simulation algorithm and its working principles are given with the thermodynamic library.

The derived model equations of multicomponent batch packed distillation column for mixtures operating at low pressures are written by FORTRAN and also in programming language of MATLAB software and they are solved with the aid of MATLAB software. The files with the extension of "xxx.m" are written using MATLAB while ones with the extension of "xxx.f" are prepared by using FORTRAN programming languages. This study is a modification of written simulation code by Yıldız (2002) for multicomponent batch plate distillation column to multicomponent batch packed distillation column.

The simulation code, which is given in Appendix C, consists of 9 "xxx.m" files for main programs, 3 "xxx.m" files for optimization and a library file written with FORTRAN programming language for thermodynamics. The m-files are "Glob\_Decs.m", "Glob\_Initial.m", "Mass\_Hydrocarbons.m", "Mass\_Polar.m", "PressureProfile.m", "Plant\_File\_Packed.m", "OptimizeR.m", "Batch\_con.m", "Batch\_obj.m" and the library file is "thermo\_LIBRARY.dll".

The library file for prediction of physical and thermodynamics properties of hydrocarbons was prepared by Dokucu (2002) and some modifications were made by Yıldız (2002). Then, Bahar (2007) modified the written equilibrium file for polar mixtures by using NRTL activity coefficient model for multicomponent mixtures. In this study, one more addition, which is the equation of NRTL activity coefficient model for binary mixtures used in experiments, is prepared and thermo library file for hydrocarbons is integrated with the file for polar

mixtures. Now, library file has two approaches ( $\phi$ - $\phi$  and  $\gamma$  -  $\phi$ ) to determine physical properties and VLE compositions for hydrocarbons and polar mixtures. The use of approach through the simulation run is determined by selecting the type of components at the start of the run.

The overall structure of the simulation code is given in Table 5.1 as in the case of Yıldız (2002).

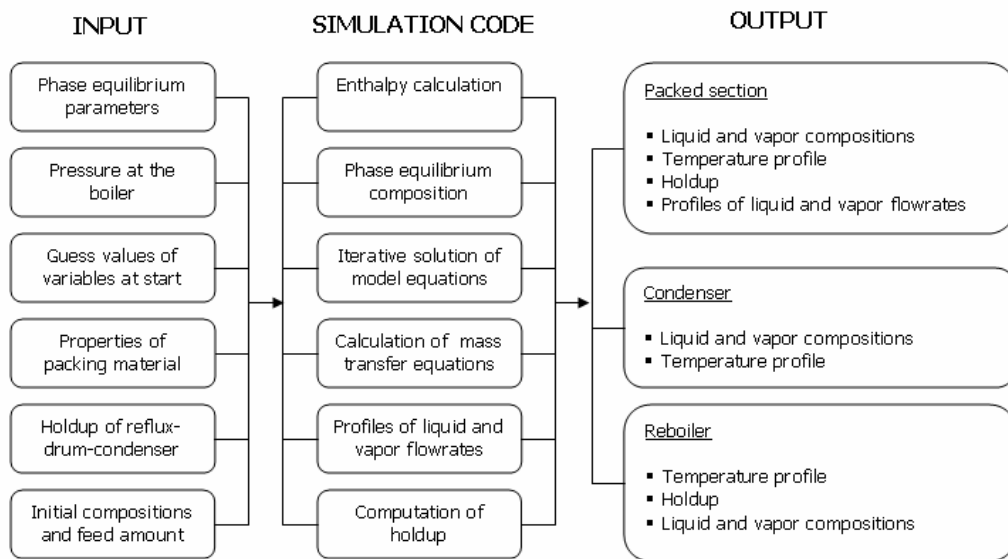
**Table 5.1** The Overall Structure of the Simulation Code.

|   |                                   |
|---|-----------------------------------|
| Main Program Codes                            | Glob_Decls.m                      |
|   | Glob_Initial.m                    |
|   | Plant_File_Packed.m               |
|   | PressureProfile.m                 |
|   | Mass_Hydrocarbons.m               |
|   | Mass_Polar.m                      |
| Thermodynamic Library<br>"thermo_LIBRARY.dll" | thermo_Init.m                     |
|   | thermo_Equilibrium_Hydrocarbons.m |
|   | thermo_Equilibrium_Polar.m        |
|   | thermo_Enthalpy.m                 |
|   | thermo_L_Density.m                |
|   | thermo_G_Density.m                |
|   | thermo_LIBRARY.f                  |
|   | thermo_LIBRARY.h                  |
|   | common_plant.h                    |
|   | parameter.h                       |
| thermo_data.dat                               |                                   |
| Optimization Code                             | OptimizeR.m                       |
|   | Batch_con.m                       |
|   | Batch_obj.m                       |

## 5.1 Main Simulation Code

As stated before, the main simulation code is a modification of the written code by Yıldız (2002) for plate distillation column. The differences between the plate distillation and packed distillation occur in hydrodynamics and in mass transfer. In plate distillation column, among the model equations there are ordinary differential equations (ODEs), however, in packed distillation if the continuous mass transfer between phases is taken into account; there are both partial differential equations (PDEs) and ODEs for the model equations. In the solution of PDEs, the approach used by Attarakih et al. (2001) is used. For the integration, a step size of  $3 \times 10^{-4}$  hours is utilized with a terminal tolerance of  $9 \times 10^{-180}$  to make the liquid and vapor component fractions zero (Yıldız, 2002).

The working principle of the simulation code for modeling is demonstrated in Figure 5.1.



**Figure 5.1** Simulation Chart of the Batch Packed Distillation Column.

"Glob\_Decs.m" and "Glob\_Initial.m" files are to initialize text files and identify parameters to be used for whole simulation algorithm. The main simulation algorithm "Plant\_File\_Packed.m" is used for determination of compositions, pressure drop, molar holdups of packed section and reboiler, flow rates and tank molar holdups, etc. Determination of pressure drop, loading and flooding points by using the approach of

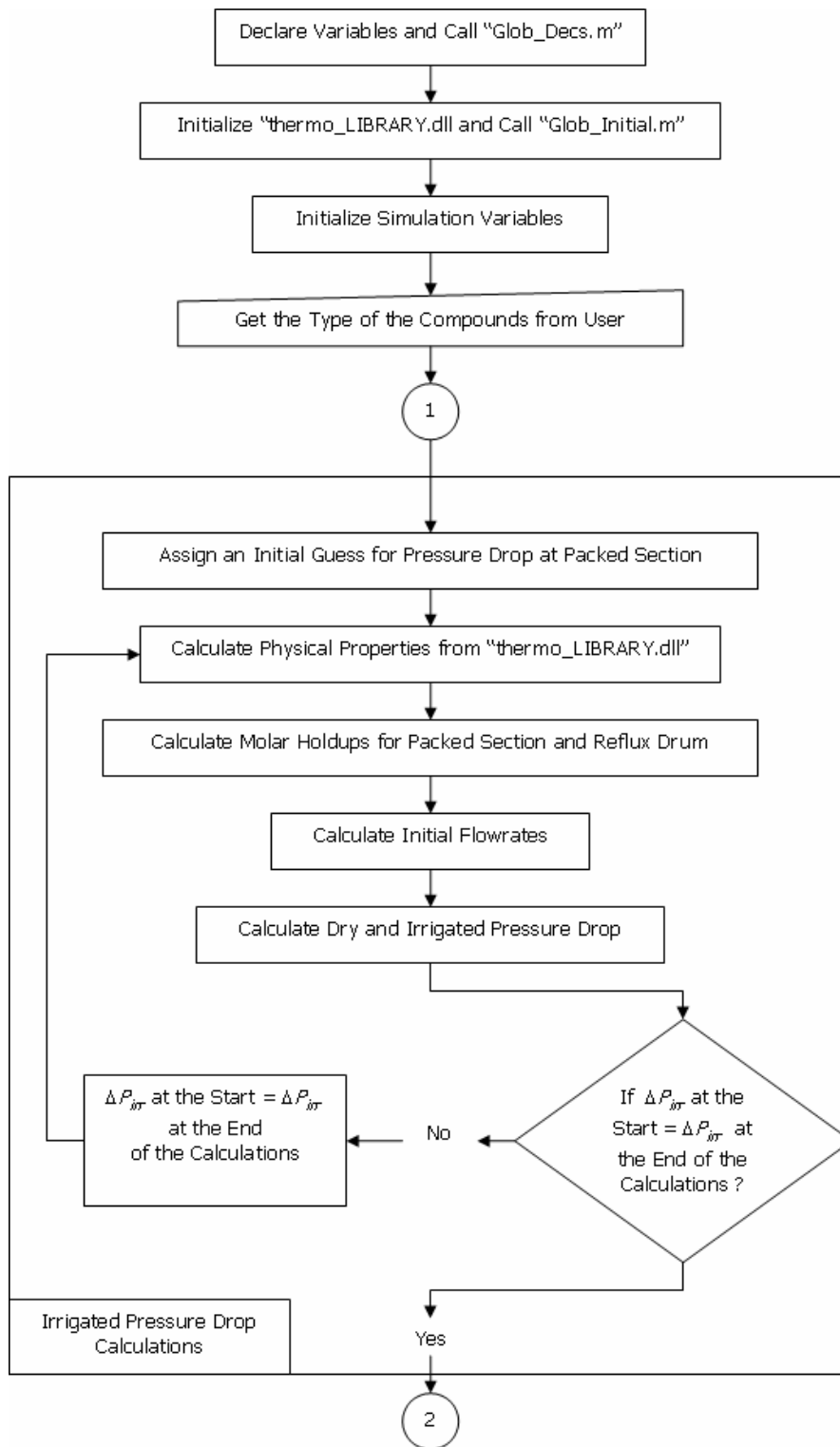
Stichlmair et al. (1989) is an iterative process. Pressure profile is assumed as linear and the computations after estimation of pressure drop are made by using "PressureProfile.m" file. The prediction of mass transfer, interfacial area correlations and physical properties are written in the files of "Mass\_Hydrocarbons.m" and "Mass\_Polar.m". The flow chart of the main simulation code is given in Figure 5.2.

## 5.2 Thermodynamic Library

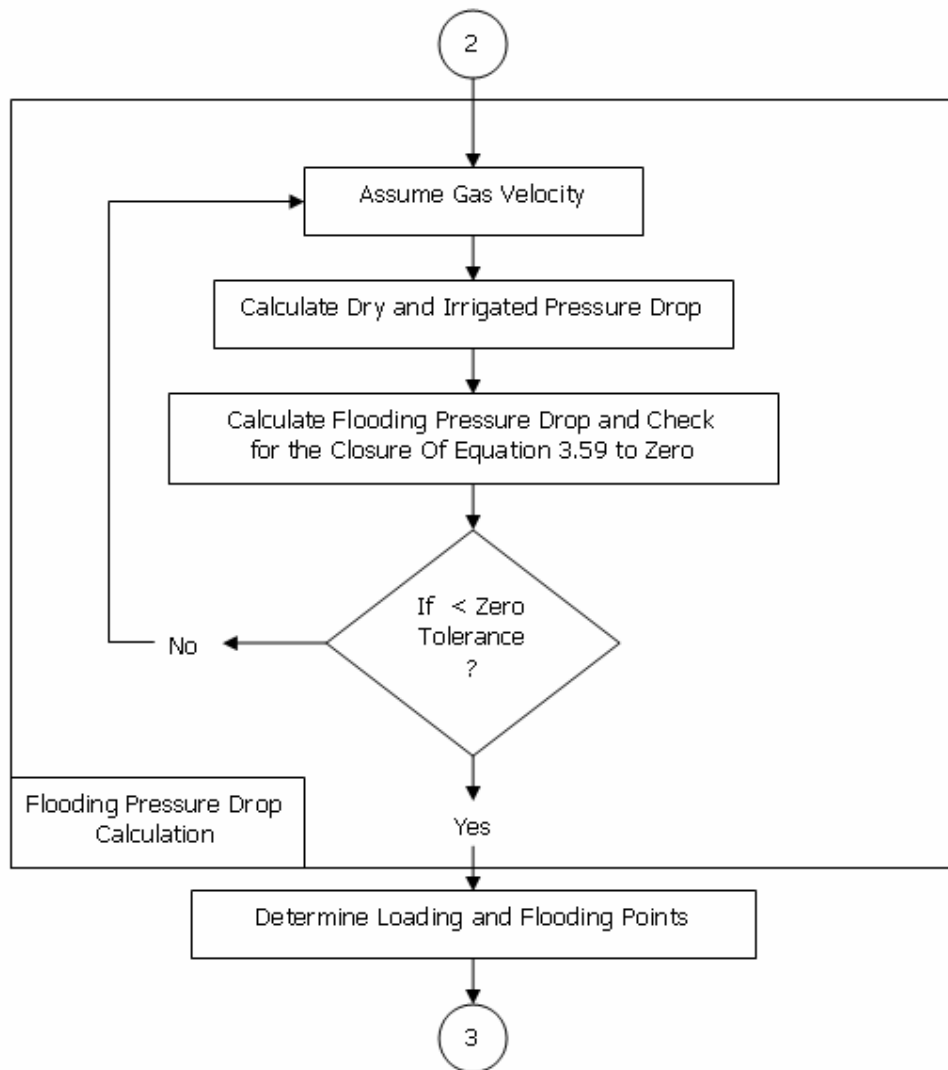
Thermodynamic library consists of files written with MATLAB and FORTRAN programming languages. The part written in FORTRAN is prepared by Dokucu (2002) for model predictive controller design of an industrial distillation column and then, modified by Yıldız (2002) to be used in MATLAB framework. Also, Bahar (2007) modified the written equilibrium subroutine to predict VLE compositions by using NRTL activity coefficient model for liquid phase and PR EOS state for vapor phase. The details of the thermodynamic library can be found from the studies of Dokucu (2002), Yıldız (2002) and Bahar (2007).

In this study, two modifications to this library are made: one is to obtain data on vapor density of the mixture besides liquid density of the mixture; the other is the addition of the equation of NRTL activity coefficient model for binary mixtures. Moreover, an integration of thermo library files written by Yıldız (2002) and Bahar (2007) is done. By using this library, one can determine enthalpy, liquid and gas density and average molecular weight of the mixture and phase equilibrium compositions by utilizing either  $\gamma - \phi$  approach or  $\phi - \phi$  approach, with the help of m-files, "thermo\_Enthalpy.m", "thermo\_Density.m", "thermo\_Equilibrium\_Hydrocarbons.m", and "thermo\_Equilibrium\_Polar.m" respectively.

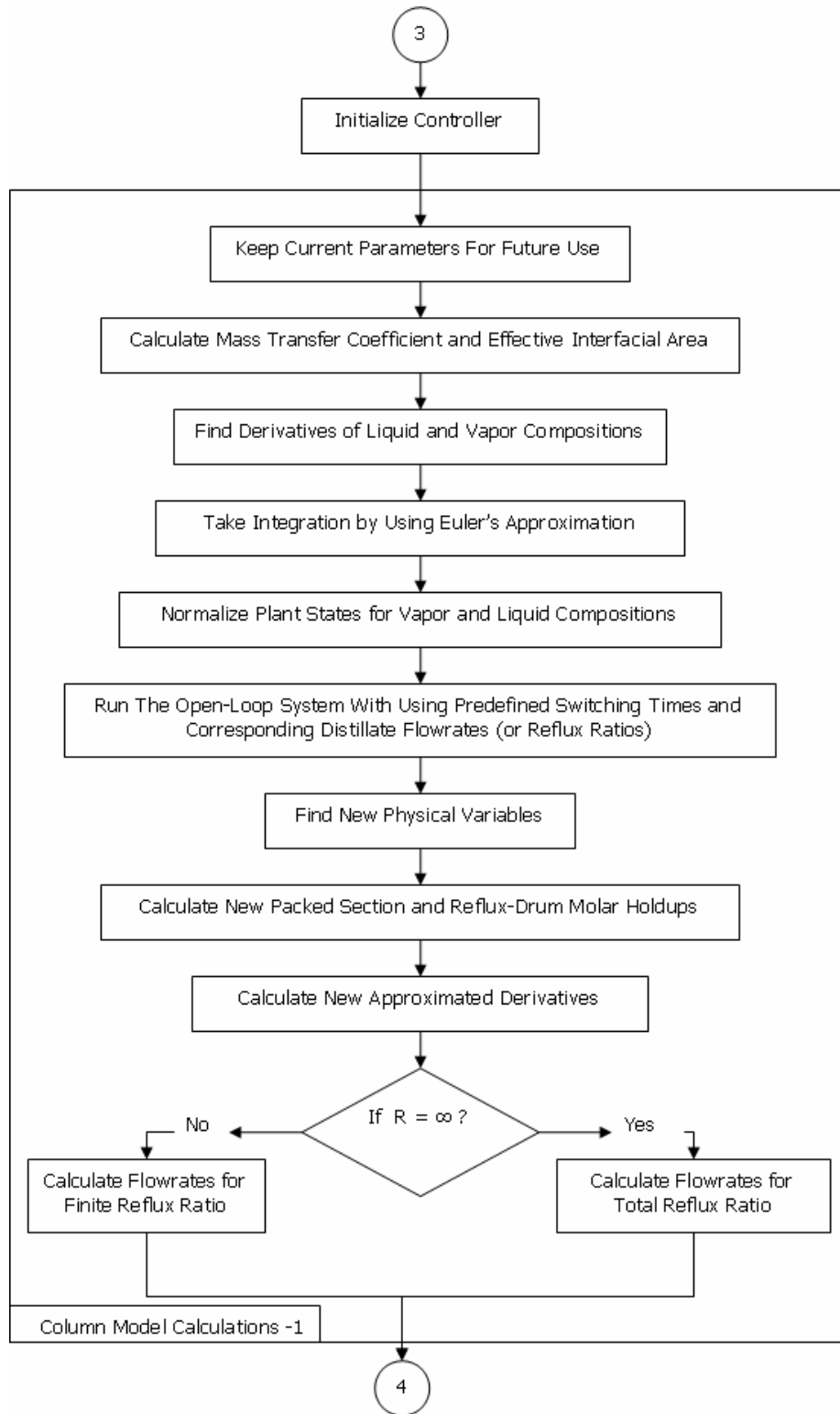




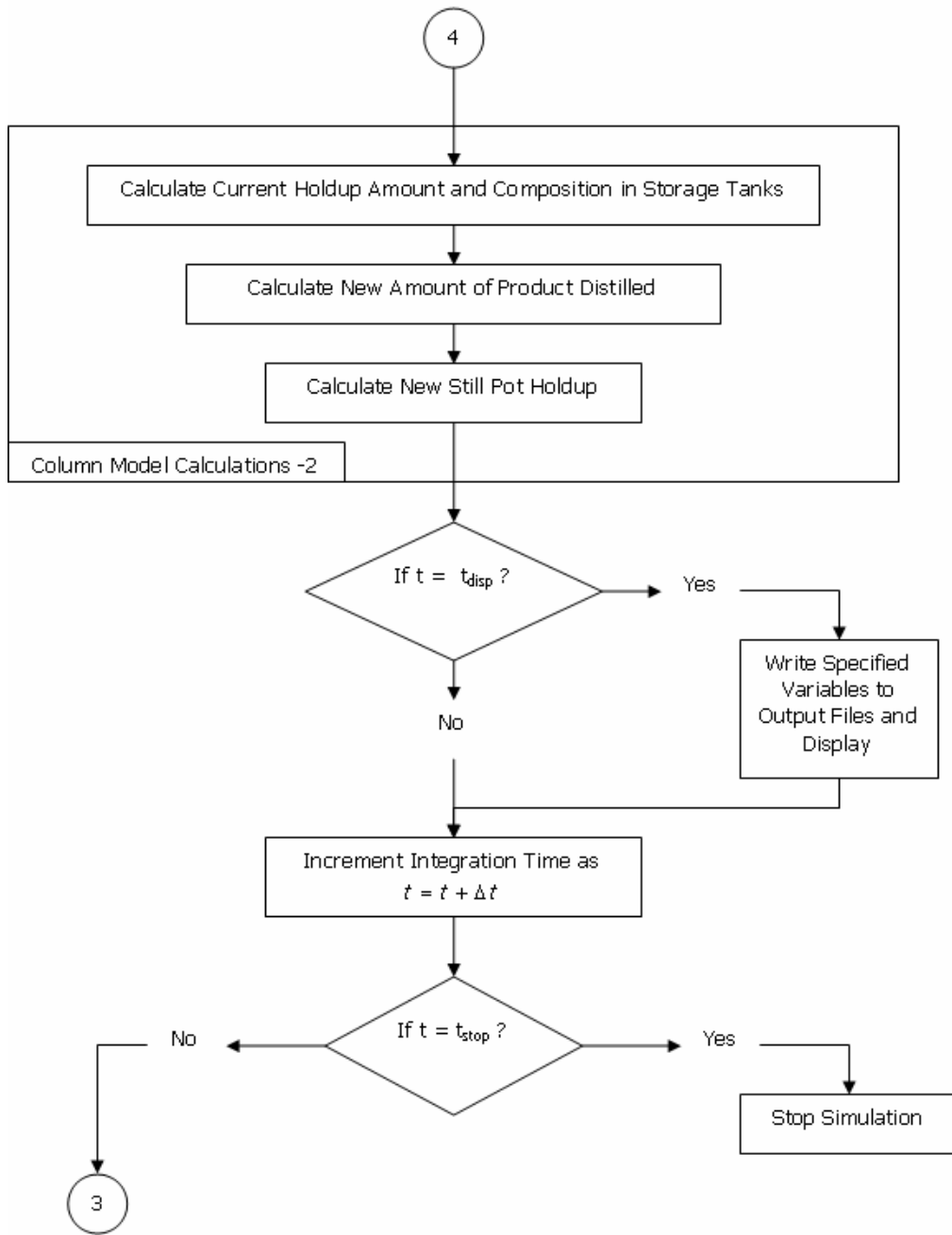
**Figure 5.2** Flow Chart of the Main Simulation Algorithm.



**Figure 5.2** Flow Chart of the Main Simulation Algorithm (Cont'd).



**Figure 5.2** Flow Chart of the Main Simulation Algorithm (Cont'd).



**Figure 5.2** Flow Chart of the Main Simulation Algorithm (Cont'd).

### 5.3 Optimization Functions

In the simulation program, the function of CAP is solved with the aid of the command "fmincon" in the MATLAB programming language (Bahar, 2007). Since the "fmincon" is to find a minimum of a constrained nonlinear multivariable function starting with an initial estimate, it is necessary to make it as a maximization function by putting a minus sign in front of the equation in the simulation algorithm. Inequality constraint to find an optimal operation point for nonlinear optimization problem is given by Equation 4.3. In the selection of optimization options in the simulation code, the ones used for medium – scale algorithms are taken into consideration. Since there is an inequality constraint, the use of large – scale algorithm is impossible. "Large scale" is chosen as "off" with termination tolerance for optimum reflux ratio of 10<sup>-3</sup>. In the medium – scale algorithms, a sequential quadratic programming (SQP) method is used. In this method, for each iteration an approximation is performed with the Hessian of the Lagrangian function by using a quasi-Newton updating method and a quadratic programming sub problem is solved. Estimation of optimum reflux ratio for an interval is done in two steps by using written simulation code by Bahar (2007). An internal reflux ratio is calculated in the first step. Iterative solutions of the written optimization code with main simulation program are generated to reach the exact solution at the second step. A global optimum internal reflux ratio to collect any product or waste material into specified tank is found out by searching global maximum points of the capacity factor for each stage by using "fmincon" command. Details of the use of "fmincon" command can be found in the study of Bahar (2007).

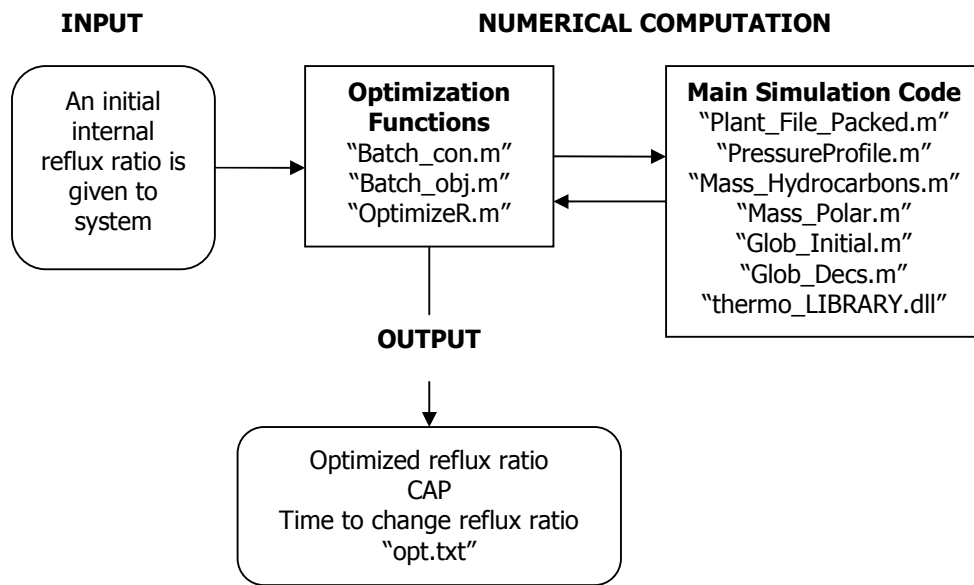
In determination of optimal reflux ratio, the main simulation code is changed to a sub function while making the optimization function as main simulation code. The procedure to determine optimal reflux ratio is as follows:

1. The amount of feed is introduced to reboiler as  $M_B$  with liquid compositions  $x_1, x_2, x_3 \dots x_j$ .
2. The boil up rate is set by tuning heat load.
3. The system is adjusted to total reflux operation conditions until the steady state composition is reached or the composition of the lightest compound is reached to desired product purity.
4. After total reflux operation, an arbitrarily initial internal reflux ratio is given to the system to start optimization. With the given reflux ratio first product is collected in the product tank until the composition in the 1<sup>st</sup> product tank is attained to specified purity level. When the composition of the 1<sup>st</sup> product tank drops below the purity level, the CAP is evaluated. Optimization program changes reflux ratio and the

simulation algorithm will start to search for another reflux ratio and the operation time will start again. This is continued until an optimum reflux ratio is obtained with terminal tolerance with  $10^{-3}$ .

- Then, obtained internal reflux ratio for the 1<sup>st</sup> product tank will be introduced to simulation algorithm and a new initial optimal reflux ratio will be given to the system to find an optimum reflux ratio for the 1<sup>st</sup> slop cut tank by guiding the distillate rate to slop cut tank. The procedure is continued until the composition of the reboiler is reached to the specified purity of the heaviest component. After several runs with assigning different values of internal reflux ratio, one which gives the maximum CAP is chosen as the optimum reflux ratio for each tank.

The flow chart of the simulation algorithm for optimization of reflux ratio profile to maximize distilled product amount is given by Figure 5.3.



**Figure 5.3** Flowchart of Simulation Algorithm for Optimum Reflux Ratio Profile.

## CHAPTER 6

### CASE COLUMN AND EXPERIMENTAL SET- UP

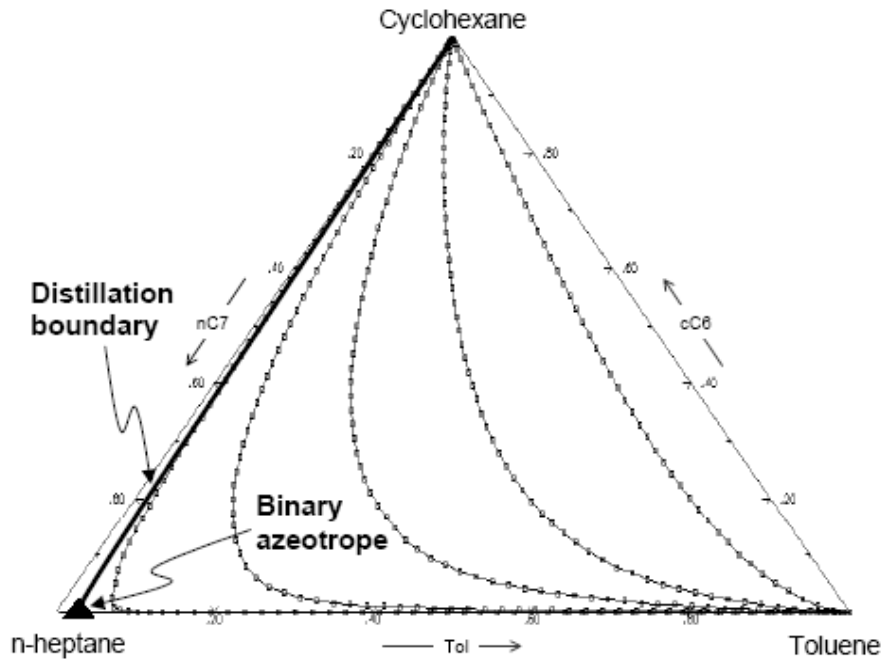
Verification of simulation code is necessary to check its validity. In this study, results of the simulation algorithm are checked for hydrocarbons and polar mixtures. The check for hydrocarbons is done analytically. Experimental studies are performed for polar mixtures, by using a binary mixture of 60% ethanol (EtOH) and 40% water (H<sub>2</sub>O). The information about components of mixtures, column specifications considered during verification of simulation code for hydrocarbons and polar mixtures, experimental procedure and analysis of the experimental findings are given below.

#### 6.1 Case Study 1: Hydrocarbons

Analysis of behavior of the simulation algorithm for hydrocarbons is performed by using a ternary mixture of cyclohexane, n-heptane and toluene. Correlations to predict physical properties of these hydrocarbons and the specifications of the components are given in Appendix D.

Residue curve map, which gives information about the phase equilibrium, can be used to trace the azeotropy of the compounds. A residue curve map of cyclohexane, n-heptane and toluene is given in Figure 6.1 (Jiménez et al., 2002).

Figure 6.1 demonstrates that “the binary system of n-heptane and toluene has a high purity binary azeotropy (0.99 mole fraction in n-heptane). This azeotropy is nonsensitive to pressure (0.975 molar to 10 atm)” (Jiménez, et al., 2002). There is not any ternary azeotropy in the system. Hence, this system can be thought as nonazeotropic system for the range of pressures used in the (up to 1 atm) simulation.



**Figure 6.1** Residue Curve Map for Cyclohexane – n - Heptane – Toluene at 1 atm (Jiménez et al., 2002).

An experimental study could not be performed with hydrocarbons due to the damage given by these solvents to the experimental set up. Therefore, accuracy of simulation algorithm will be performed qualitatively as in the case of Hitch et al. (1988) by comparing nine simulation runs with varying one parameter at each simulation run. Case column is considered as having the same specifications of the column and feed used in the study of Mujtaba and Macchietto (1993). Column and feed specifications are given in Tables 6.1.

The column specifications in the study of Mujtaba and Macchietto (1993) are given as tray numbers, boil up rate, holdups of the trays, holdups of condenser-reflux-drum and pressure drop throughout the column. Since the column used in the study of Mujtaba and Machietto (1993) is a plate distillation column, some more specifications has to be included as inputs to the simulation program for the packed distillation column. Thus, irrigated and flooding pressure drop is evaluated by the simulation code using the top pressure given in Table 6.1. Also, the design gas rate is selected to be 38% of the flooding rate for a column of 0.35m in



diameter with packed height of 1.5m. The calculated specifications of the column and specifications of the used packing material are given in Tables 6.2 and 6.3, respectively.

**Table 6.1** Column and Feed Specifications.

|  |        |
|--|--------|
| Condenser – Reflux- Drum Holdup (kmol) | 0.02   |
| Top Pressure (Pa)                      | 101600 |
| Maximum Boil up Rate (kmol/h)          | 2.75   |
| Fresh Feed Amount (mol)                | 2930   |
| Initial mole fractions at reboiler     |        |
| Cyclohexane                            | 0.407  |
| n- Heptane                             | 0.394  |
| Toluene                                | 0.199  |
| Desired purity of the first product    | 0.9    |
| Desired purity of the second product   | 0.8    |

**Table 6.2** Additional Specifications for the Packed Distillation Column Used in the Simulation.

| <b>Parameter</b>                        | <b>Value</b> |
|---|--------------|
| Irrigated pressure drop (Pa)            | 359.52       |
| Flooding pressure drop (Pa)             | 2637.23      |
| Design gas rate / Flooding Gas Rate (%) | 0.38         |
| Number of differential element, NT      | 40           |
| Diameter of the column (m)              | 0.35         |
| Height of packed section (m)            | 1.5          |

**Table 6.3** Specifications of Used Random Packing Material.

| <b>Parameter</b>                                  | <b>Value</b>         |
|---|----------------------|
| Packing material                                  | Ceramic Raschig ring |
| Diameter(m)                                       | 0.010                |
| Specific surface area of the packing ( $m^{-1}$ ) | 472                  |
| Porosity  | 0.655                |

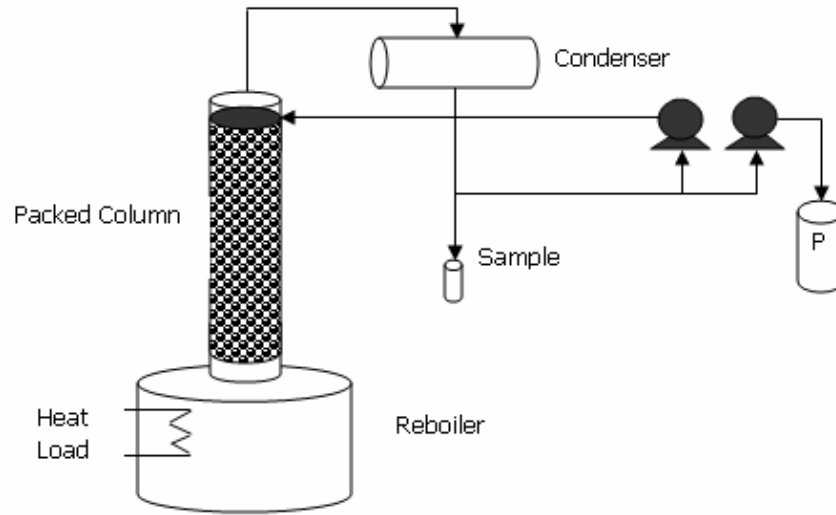
The heat load in the selected distillation column is kept at the maximum value and internal reflux ratio is used as the manipulated variable to understand the optimum operation to maximize distillate product amount, since it is assumed that there is a perfect level control at reflux –drum – condenser section (Yıldız, 2002).

## **6.2 Case Study 2: Polar mixtures**

The accuracy of simulation algorithm is verified quantitatively by carrying out experiments on a lab scale packed distillation column by using 60% EtOH and 40% H<sub>2</sub>O mixture. Ethanol of  $\geq 99.99\%$  (w/w) purity (supplied by Merck) and pure water are used to prepare the mixture. Physical properties for polar mixtures assumed as constant in the simulation code in order to simplify the complex calculations. Physical properties and the specifications of ethanol and water are given in Appendix D. Azeotropy of ethanol and water mixture occurs at 78.2 °C at 1 atm (4.4% by weight water and 95.6% by weight ethanol).

### **6.2.1 Experimental Set – Up**

The experimental set – up used in this study (Figure 6.2) consists of a packed distillation column made of glass and with an ID of 5cm and with 0.4m packing height. Column is filled with 5 mm diameter glass Raschig Rings. The polar mixture to be analyzed is 1000 ml solution with 60% EtOH and 40% H<sub>2</sub>O [mol/mol]. The mixture is initially 23 moles. Heat load is chosen as 500 W (1800000 J/h). Flow rate of condenser is 1.4 l/min. In order to reduce heat loss to the environment, reboiler and packed column are insulated. Peristaltic pumps are used to pump distilled material to column and product cut tanks. Calibration curve of peristaltic pumps is given in Appendix E.



**Figure 6.2** Schematic View of Experimental Set – Up.

### 6.2.2 Experimental Procedure

Analyses of the distillate samples taken in the experiments are done in the setup explained above are measured by using Porapak - Q column in gas chromatography (GC) (Hewlett – Packard 5890 Series II) by using nitrogen as carrier gas. First, a calibration curve is prepared with the samples of known compositions. Details of this study and calibration equations are given in Appendix E. Then, reboiler is filled with 1000 ml solution with 60% EtOH and 40% H<sub>2</sub>O and the heat load is adjusted to 0.5 kW. The column is operated at total reflux for about five hours - until the steady state is reached. The operation is continued with a finite external reflux ratio of 0.5 for one hour. Contents of the reboiler is started to boil in 15 minutes after start and when the condensed vapor is observed at the reflux- drum, samples are taken from the distilled product in every 5 minutes for about 0.5 hrs to observe the rise of the ethanol composition at the reflux drum. Then, samples are taken after 0.5 hrs in every 20 minutes, until total reflux period ends and again in every 5 minutes until distillation stops. Then, samples are analyzed in GC. The liquid compositions of ethanol are obtained with the use of the following equation:

$$1.0152\kappa_{H_2O}x_{EtOH}^3 - 2.1915\kappa_{H_2O}^2 + (\kappa_{EtOH} + 1.9308\kappa_{H_2O})x_{EtOH} - \kappa_{EtOH} = 0 \quad (6.1)$$

where,  $\kappa_{H_2O}$  and  $\kappa_{EtOH}$  are area of the water and ethanol obtained from GC, and  $x_{EtOH}$  is the liquid composition of ethanol [mol/mol].

## CHAPTER 7

### RESULTS AND DISCUSSION

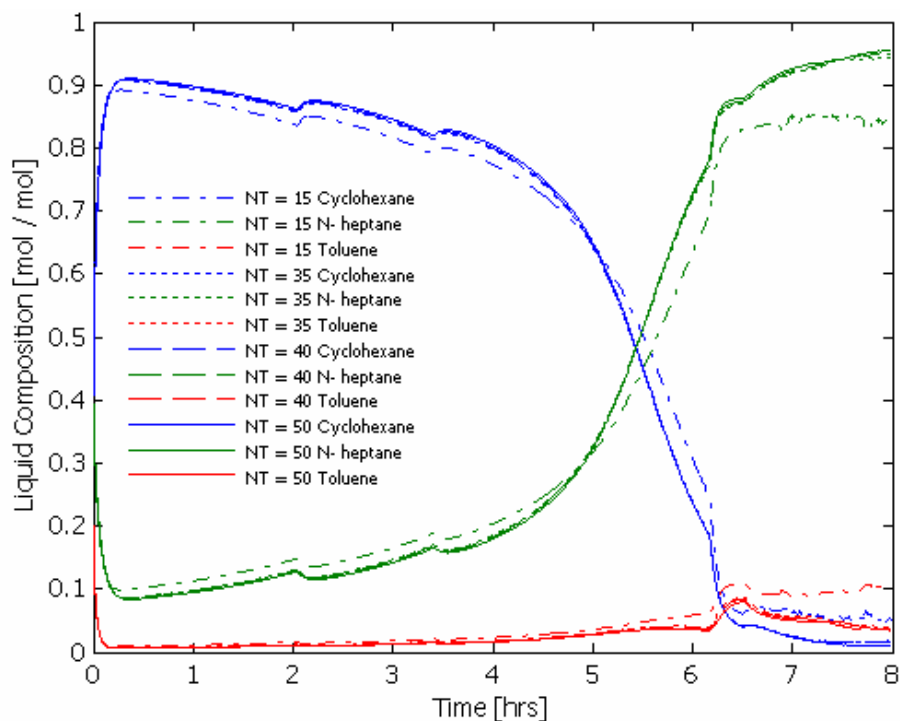
Verification of the simulation code for the separation of hydrocarbon and polar mixtures using different simulation runs and experimental findings, optimum reflux ratio profile for maximizing the amount of distillate in a specific time and recycling of molar holdups of slop cut tank for the proceeding batch operation are given and discussed in this chapter.

#### 7.1 Effect of Incremental Bed Height on the Results of the Simulation

The selection of the size of the incremental packing bed height of the batch packed distillation column,  $\Delta z$  ( $\Delta z = \text{Bed Height (Z)} / \text{Number of Incremental Element (NT)}$ ) is important for the accuracy of the results. Thus, the effect of NT (15, 35, 40 and 50) on the accuracy of the results is investigated first by using related study (Yıldız (2002)) found from the literature by using an internal reflux ratio profile given in Table 7.1 . It is observed from Figure 7.1 that, incremental element,  $\Delta z$ , has an effect on the separation operation as separation becomes more difficult (after 6 hrs of operation, separation is between n-heptane and toluene and is more difficult than the separation of cyclohexane and n-heptane). As the number of incremental elements increases, separation efficiency increases. However, the effect of number of incremental element on separation efficiency levels off as the number reaches 50. That is, there is a slight change between the results of simulation code for NT = 40 and NT = 50. Therefore, during the simulation runs in the following sections, NT is taken as 40.

**Table 7.1** Internal Reflux Ratio Profile of the Study of Yıldız (2002).

| Time Interval [hrs] | Internal Reflux Ratio (L / V) |
|---------------------|-------------------------------|
| 0 – 2.04            | 0.875                         |
| 2.04 – 3.4          | 0.911                         |
| 3.4 – 6.17          | 0.933                         |
| 6.17 – 6.51         | 0.831                         |
| 6.51 – 8.00         | 0.876                         |



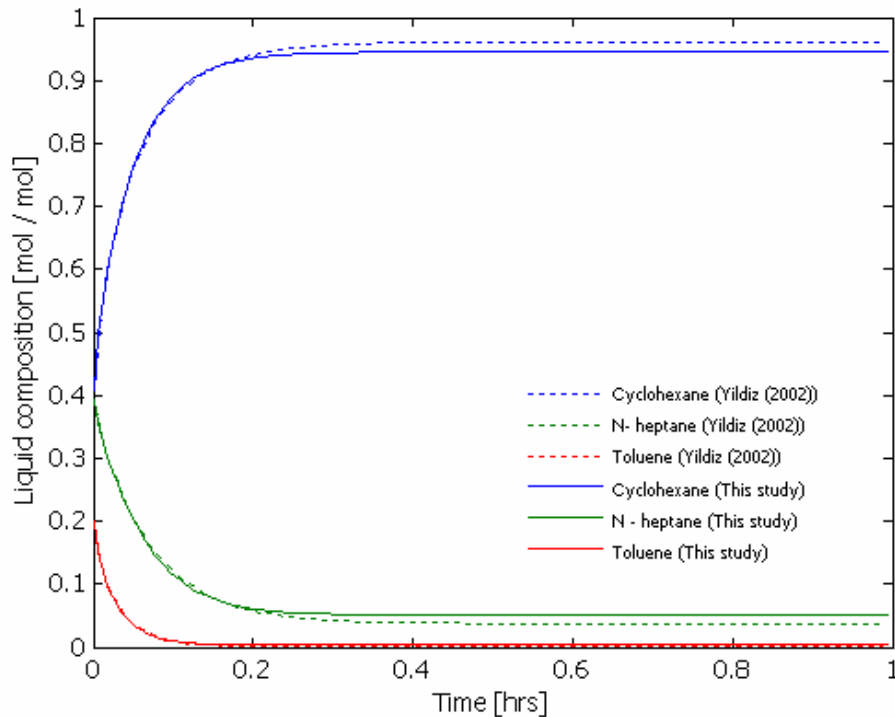
**Figure 7.1** Comparison of Distillate Compositions for Different Incremental Elements.

## 7.2 Verification of Simulation Code by the Study of Yıldız (2002)

The studies on the verification of the simulation code are firstly performed by comparing the distillate compositions of the simulation results of the Yıldız (2002) in a plate distillation column: at total reflux, at a constant internal reflux ratio of 0.875 and at the piecewise constant internal reflux ratio profile as a function of time given in Table 7.1 (Yıldız (2002)). This validation is done as a starting point to check whether the results of the simulation

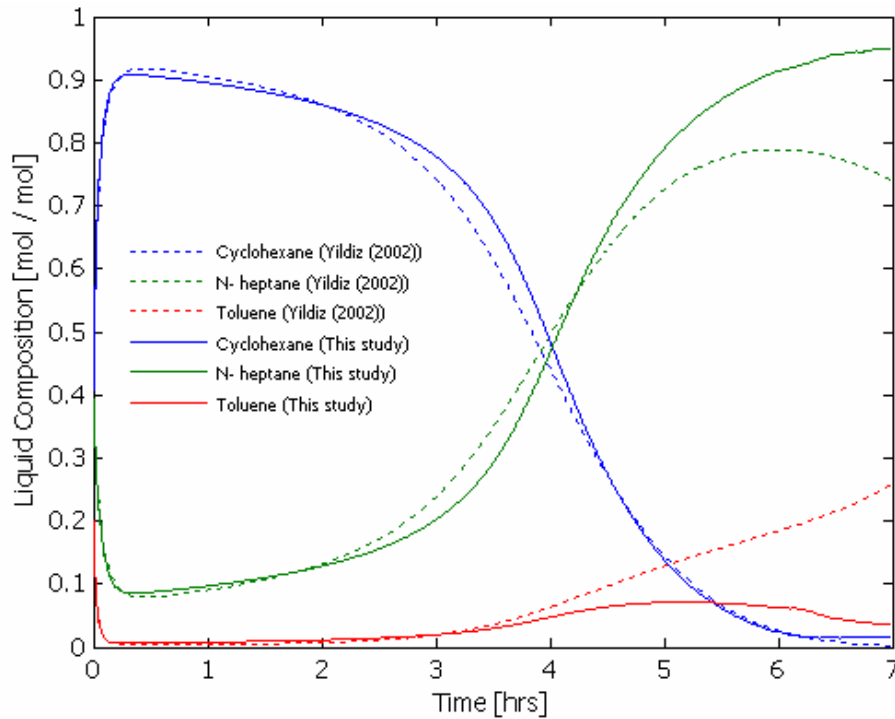
code are reasonable or not. After this section, verification of the simulation code will be done theoretically and experimentally.

Although the study of Yildiz (2002) is done in a batch plate distillation column with 20 trays (including reboiler and condenser) for a multicomponent mixture, the distillate compositions can be compared with the dynamic behavior of the multicomponent batch packed distillation column which has the same feed amount, condenser hold-up and heat input. Of course depending upon L/D ratio for the column and packing material, the separation efficiency will be different. Nevertheless the dynamic trend is wanted to be checked in terms of distillate compositions. In Figure 7.2, the comparison of the distillate compositions between the results of simulation code prepared by Yildiz (2002) and those of simulation code written in this study for total reflux period of one hour is given. It is observed that, steady state distillate compositions for plate distillation column are slightly higher than those for packed distillation column. However, the dynamic behavior of the composition of the components for packed distillation is in a similar trend with the plate distillation column and the time necessary to reach steady state is nearly the same for both cases. The difference in distillate composition shown in Figure 7.2 can be reduced by increasing the packing height.



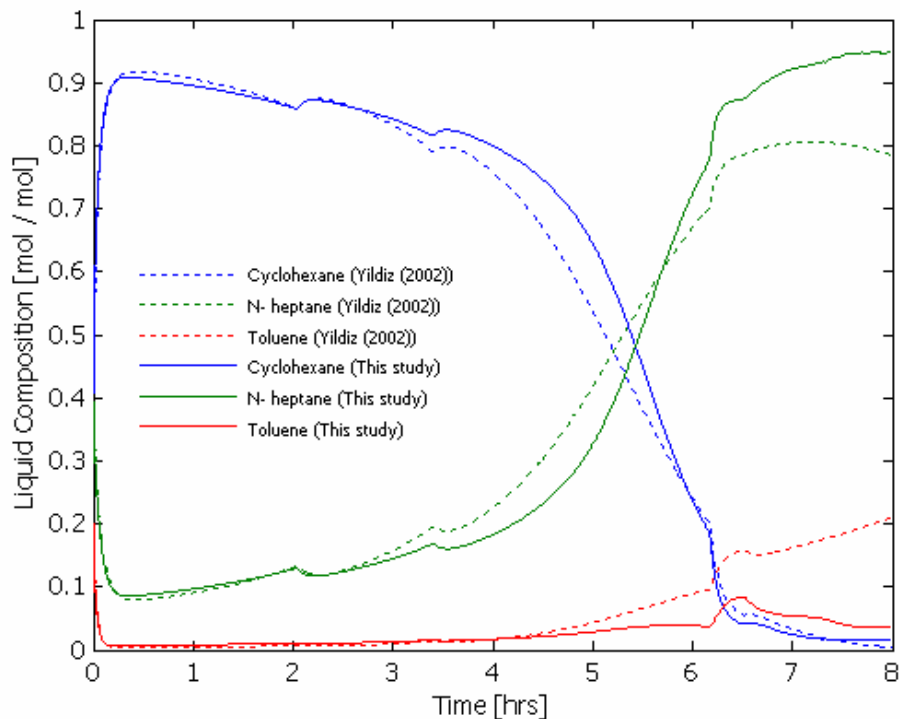
**Figure 7.2** Comparison of Distillate Compositions at Total Reflux.

In Figure 7.3, simulation run performed for a constant internal reflux ratio of 0.875 is given. Again, there is a good agreement between the dynamic behaviors of the distillate compositions of two studies. In packed distillation column, the separation efficiency is higher for the second lightest compound, n- heptane. Also, intersection point of the distillate compositions of cyclohexane and n-heptane is nearly the same for both columns.



**Figure 7.3** Comparison of Distillate Compositions at Internal Reflux Ratio of 0.875.

In the study of Yıldız (2002) reflux ratio profile (Table 7.1 ) previously evaluated and used by Mujtaba and Macchietto (1993) was considered. Using this profile simulation results are obtained for the dynamic behavior of the distillate compositions and are shown Figure 7.4. As can be seen from the figure, the separation efficiency of the packed distillation column for the second lightest compound is again higher than that of the plate distillation column and intersection point of the lightest compound (cyclohexane) and the second lightest compound (n-heptane) is approximately 5.1 hrs for the plate distillation column and 5.43 hrs for the packed distillation column. Thus, the trend of the distillate compositions is again similar for two distillation columns. Thus, the simulation code written in this study for the batch packed distillation column is verified in trend to a certain extent.



**Figure 7.4** Comparison of Distillate Compositions with a Specified Reflux Ratio Profile.

### 7.3 Verification of Simulation Code for Hydrocarbon Mixtures

Hitch and Rousseau (1988) has prepared a simulation code for a continuous packed distillation column with random packing which was used to separate a multicomponent system (propane, n-butane, n-hexane) at high pressures. They checked the validation of code by performing runs to see the expected effects of different variables on the distillate compositions.

In the hydrocarbon system studied, experiments could not be performed in the experimental column system due to the high etching effect of the mixture on column joints made of different plastics. Therefore, a similar approach to Hitch and Rousseau (1988) is used for the verification of the code for hydrocarbon mixtures.

In batch distillation operation, there are two important periods: one is the startup period and the other is the production period. In startup period, the column is operated at total reflux until the composition at the condenser reaches a constant value. In the production period, the composition in the condenser changes since distillate is withdrawn from the column. The



simulation program written in this study can handle both – startup and production periods, and solves the state and algebraic equations to find the vapor and liquid compositions, change of temperatures, holdups, liquid and flow rates during operation time for startup and production periods. In the simulation runs, the column is first operated at total reflux for about one hour and then, production period is started. The column reaches steady state in one hour at total reflux.

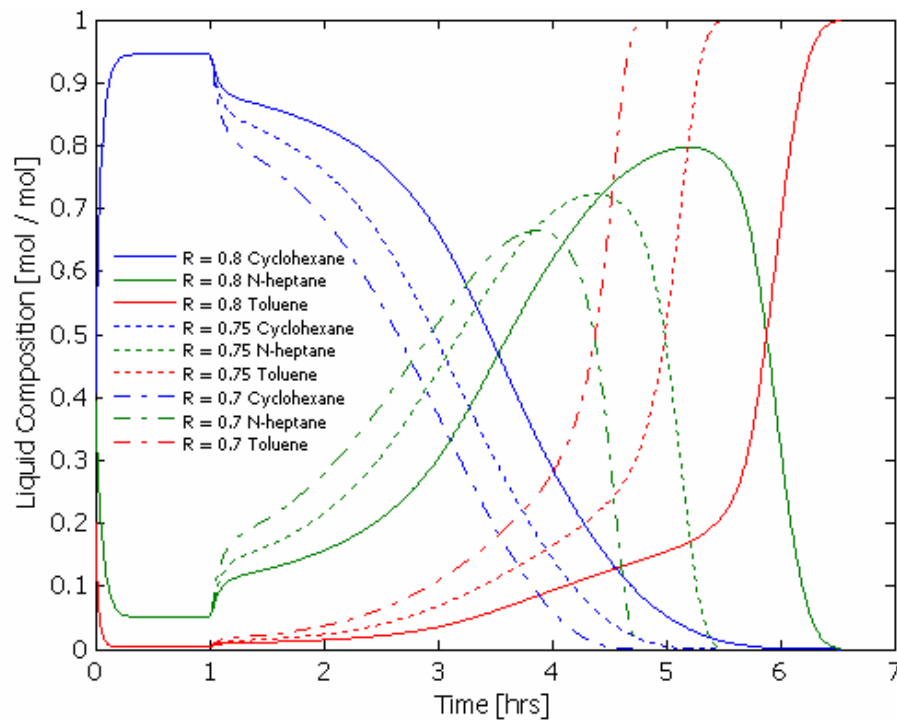
In order to test the simulation code prepared, nine simulation runs are done similar to the study done by Hitch and Rousseau (1988). During these runs, internal reflux ratio, height of packing, condenser holdup and heat load given to the system are changed and the effect of these variables on distillate compositions are analyzed and checked whether expected behaviors are achieved or not. In Table 7.2 the values of the parameters that are used in simulation runs are given. First run is chosen as the base run.

**Table 7.2** Parameters Used in Different Simulation Runs.

| <b>Run Number</b> | <b>Height of Packing, m</b> | <b>Internal Reflux Ratio (L/V)</b> | <b>Condenser Holdup, mol</b> | <b>Heat Load, J/hr</b>             |
|-------------------|-----------------------------|------------------------------------|------------------------------|------------------------------------|
| 1                 | 1.5                         | <b>0.8</b>                         | 20                           | $8.15 \cdot 10^7$                  |
| 2                 | 1.5                         | <b>0.75</b>                        | 20                           | $8.15 \cdot 10^7$                  |
| 3                 | 1.5                         | <b>0.7</b>                         | 20                           | $8.15 \cdot 10^7$                  |
| 4                 | <b>2</b>                    | 0.8                                | 20                           | $8.15 \cdot 10^7$                  |
| 5                 | <b>3</b>                    | 0.8                                | 20                           | $8.15 \cdot 10^7$                  |
| 6                 | 1.5                         | 0.8                                | <b>50</b>                    | $8.15 \cdot 10^7$                  |
| 7                 | 1.5                         | 0.8                                | <b>80</b>                    | $8.15 \cdot 10^7$                  |
| 8                 | 1.5                         | 0.8                                | 20                           | <b><math>6.5 \cdot 10^7</math></b> |
| 9                 | 1.5                         | 0.8                                | 20                           | <b><math>5 \cdot 10^7</math></b>   |

The column is operated at constant internal reflux ratios of 0.8, 0.75 and 0.7 at the production period for simulation runs 1, 2 and 3, respectively. Figure 7.5 demonstrates the effect of internal reflux ratio on distillate compositions. As seen from the Figure 7.5 the highest purity of cyclohexane is obtained after one hour of total reflux period. Steady state distillate compositions at total reflux and distillate compositions for first, second and third simulation runs at 2 hr elapsed time are given in Table 7.3. Since at total reflux all three

cases have same reflux ratio the distillate compositions are equal to each other. On the other hand, the distillate purity in cyclohexane decreases as reflux ratio decreases at production period. There are 8.18% and 17.45% decrease for compositions of cyclohexane of simulation run 2 and 3 with respect to distillate compositions of simulation run 1, respectively for the same time elapsed. Moreover, there are 37.59% and 78.05% increase at the distillate compositions of n – heptane as reflux ratio decreases. For toluene, liquid compositions also increase and, changes of compositions of simulation run 2 and 3 with respect to distillate compositions of simulation run 1 are 60.14% and 150.68%, respectively. Furthermore, in terms of n-heptane, it is observed that, maximum concentration obtained is decreasing as reflux ratio decreases. Consequently, the time necessary to complete removal of the heaviest compound from the reboiler increases, as the reflux ratio becomes closer to total reflux value. Operation times are found to be 6.54, 5.478 and 4.74 hrs for the reflux ratios with descending order, respectively. This trend is an expected one in a batch distillation system as found also by Hitch and Rousseau (1988).



**Figure 7.5** Effect of Internal Reflux Ratio on the Distillate Compositions.

**Table 7.3** Distillate Compositions at Different Reflux Ratios at Two Hours Elapsed Time.

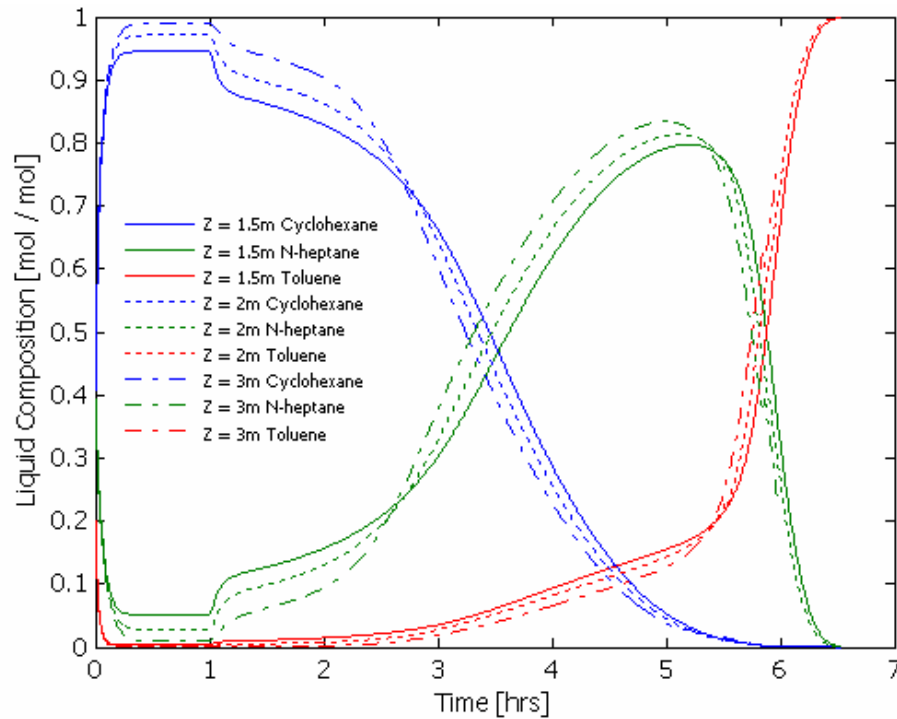
| Time [hrs] | Internal Reflux | Liquid Compositions, mole fractions |            |         |
|------------|-----------------|-------------------------------------|------------|---------|
|            | Ratio           | Cyclohexane                         | N- heptane | Toluene |
| 2.00       | 0.8             | 0.8285                              | 0.1567     | 0.0148  |
| 2.00       | 0.75            | 0.7607                              | 0.2156     | 0.0237  |
| 2.00       | 0.7             | 0.6839                              | 0.2790     | 0.0371  |

**Table 7.4** Percentage Changes of Distillate Compositions at Different Internal Reflux Ratios.

| Compound    | Internal Reflux Ratio | Percentage Change (%) |
|-------------|-----------------------|-----------------------|
| Cyclohexane | 0.8                   | -                     |
|             | 0.75                  | -8.18                 |
|             | 0.7                   | -17.45                |
| N – heptane | 0.8                   | -                     |
|             | 0.75                  | 37.59                 |
|             | 0.7                   | 78.05                 |
| Toluene     | 0.8                   | -                     |
|             | 0.75                  | 60.14                 |
|             | 0.7                   | 150.68                |

In the second phase, height of packing is changed from 1.5m to 2m and 3m to observe its effect on separation, while keeping other parameters constant at base run (internal reflux ratio = 0.8, condenser holdup = 20 moles and heat load =  $8.15 \times 10^7$  J/h). The effect of changing height of packing on distillate compositions is shown in Figure 7.6. The total effective interfacial area for mass transfer increases as the height of packing increases which increases the total mass transferred. It is observed from Figure 7.6 that, the maximum concentrations of the lightest (in total reflux period) and second lightest components (in production period) increases as packed height increases as expected. Distillate compositions are different from the compositions of base simulation run not only at finite reflux ratio but also at total reflux ratio as given in Tables 7.5 and 7.6. There are 2.73% and 4.84% increase at the distillate compositions of cyclohexane and, there are 45.20% and 82.27% decrease for distillate compositions of n- heptane at total reflux as height of packed section increases. Changes of percentages of distillate compositions are increased for cyclohexane

and decreased for n-heptane and toluene since separation efficiency of the column increases as the height of packed section increases. Moreover, the total operation time is found to be decreasing slightly 6.54, 6.52 and 5.976 hrs as packing height increases from 1m to 3m. This also verifies the code.



**Figure 7.6** Effect of Height of Packings on the Distillate Compositions.

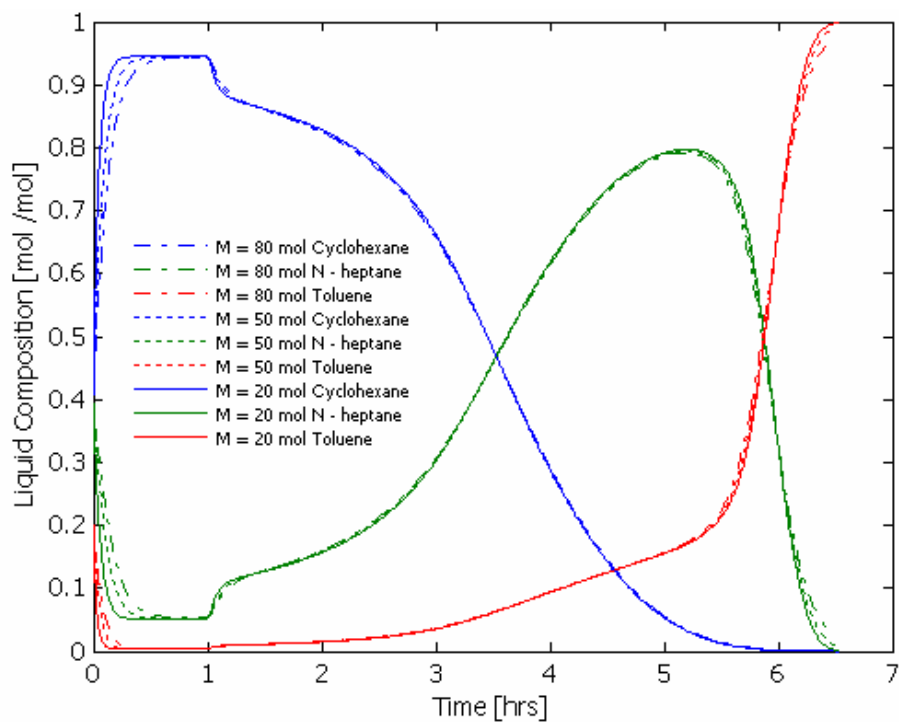
**Table 7.5** Distillate Compositions at Different Height of Packings at Two Hours Elapsed Time.

| Time [hrs] | Height of Packings [m] | Liquid Compositions, mole fractions |            |         |
|------------|------------------------|-------------------------------------|------------|---------|
|            |                        | Cyclohexane                         | N- heptane | Toluene |
| 1.00       | 1.5                    | 0.9451                              | 0.0513     | 0.0036  |
| 1.00       | 2                      | 0.9709                              | 0.0280     | 0.0011  |
| 1.00       | 3                      | 0.9908                              | 0.0091     | 0.0001  |
| 2.00       | 1.5                    | 0.8285                              | 0.1567     | 0.0148  |
| 2.00       | 2                      | 0.8621                              | 0.1301     | 0.0078  |
| 2.00       | 3                      | 0.9031                              | 0.0945     | 0.0024  |

**Table 7.6** Percentage Change of Distillate Compositions at Different Height of Packings.

| Compound    | Height of Packings | Percentage Change (%) |              |
|-------------|--------------------|-----------------------|--------------|
|             |                    | Time = 1 hrs          | Time = 2 hrs |
| Cyclohexane | 1.5                | -                     | -            |
|             | 2                  | 2.73                  | 4.06         |
|             | 3                  | 4.84                  | 9.00         |
| N – heptane | 1.5                | -                     | -            |
|             | 2                  | -45.20                | -16.98       |
|             | 3                  | -82.27                | -39.7        |
| Toluene     | 1.5                | -                     | -            |
|             | 2                  | -69.44                | -47.30       |
|             | 3                  | -97.22                | -83.78       |

Effect of condenser holdups on separation is demonstrated in Figure 7.7. Condenser holdup is increased from 20 moles with 30 moles increments up to 80 moles while keeping other parameters constant at the base run (height of packing = 1.5m, internal reflux ratio = 0.8 and heat load =  $8.15 \times 10^7$  J/h). It is observed that, the effect of condenser holdup is only seen during the startup period. The time necessary to reach steady state increases as the molar holdups of the condenser increases as expected. Although, there is a slight difference in the distillate compositions with time at the startup period due to different condenser holdups, steady state compositions at the end of the total reflux period are nearly same for the three runs as given in Tables 7.7 and 7.8. When given information at tables is analyzed, the greatest change in distillate compositions occurs as 5.56% at toluene at the end of the startup period with 80 mol condenser holdup. On the other hand, there is a slight effect of varying molar holdups of condenser on distillate compositions during the production period. The operation times and obtained product purities of compounds are almost the same for the three simulation runs since the amount of fresh feed is much greater than the amount of molar holdup of condenser.



**Figure 7.7** Effect of Condenser Holdups on the Distillate Compositions.

**Table 7.7** Distillate Compositions at Different Condenser Holdups for Two Hours Elapsed Time.

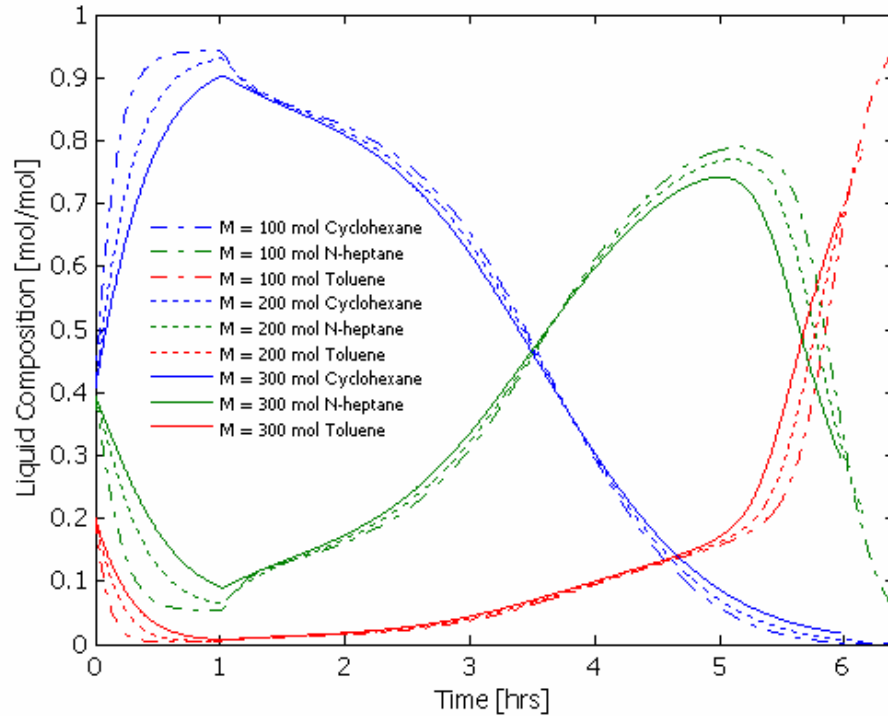
| Time [hrs] | Condenser Holdup [mol] | Liquid Compositions, mole fractions |            |         |
|------------|------------------------|-------------------------------------|------------|---------|
|            |                        | Cyclohexane                         | N- heptane | Toluene |
| 1.00       | 20                     | 0.9451                              | 0.0513     | 0.0036  |
| 1.00       | 50                     | 0.9442                              | 0.0521     | 0.0037  |
| 1.00       | 80                     | 0.9432                              | 0.0530     | 0.0038  |
| 2.00       | 20                     | 0.8285                              | 0.1567     | 0.0148  |
| 2.00       | 50                     | 0.8265                              | 0.1584     | 0.0151  |
| 2.00       | 80                     | 0.8242                              | 0.1603     | 0.0155  |

**Table 7.8** Percentage Change of Distillate Compositions at Different Condenser Holdups.

| Compound    | Condenser Holdups [mol] | Percentage Change (%) |             |
|-------------|-------------------------|-----------------------|-------------|
|             |                         | Time = 1 hr           | Time = 2 hr |
| Cyclohexane | 20                      | -                     | -           |
|             | 50                      | -0.10                 | -0.24       |
|             | 80                      | -0.20                 | -0.52       |
| N – heptane | 20                      | -                     | -           |
|             | 50                      | 1.56                  | 1.09        |
|             | 80                      | 3.31                  | 2.30        |
| Toluene     | 20                      | -                     | -           |
|             | 50                      | 2.78                  | 2.03        |
|             | 80                      | 5.56                  | 4.73        |

In the study of Hitch and Rousseau (1988) the condenser holdup is comparable with the fresh feed which resulted in change of composition profiles with condenser holdup. In this study, condenser holdups of simulation run 1, 6 and 7 are too small than the total feed. Therefore, condenser holdup is increased from 100 moles with 100 moles increments up to 300 moles to observe the effect of condenser holdup during production period while keeping other parameters constant at the base run (height of packing = 1.5m, internal reflux ratio = 0.8 and heat load =  $8.15 \times 10^7$  J/h). It is observed from Figure 7.8 that the steady state compositions are not reached for the simulation runs with 200 and 300 moles of condenser holdup during one hour startup period. Hence, the startup period, which is necessary to reach a steady state value, is increased from one hour to 2.5 hours. It is observed from Figure 7.9 that all simulation runs reach a steady state value in 2.5 hours. As it is stated before, the time necessary to reach a steady state value increases as the amount of condenser holdup increases. Steady state compositions given in Table 7.9 are nearly same for 100, 200 and 300 moles condenser holdups. There are 0.36% and 0.80% decrease on steady state compositions of cyclohexane, 5.79% and 12.90% increase on steady state compositions of n-heptane and 7.90% and 15.79% increase on steady state compositions of toluene as the amount of condenser holdup increases at startup period as stated in Table 7.10. In product period, the separation efficiency of column decreases as the amount of condenser holdup increases. The percentage changes of distillate compositions of cyclohexane and toluene in production period when they are compared with these of startup period. However, this is not the case for distillate compositions of n-heptane at production

period. The maximum purity of n-heptane is obtained when the amount of condenser holdup is equal to 100 moles.

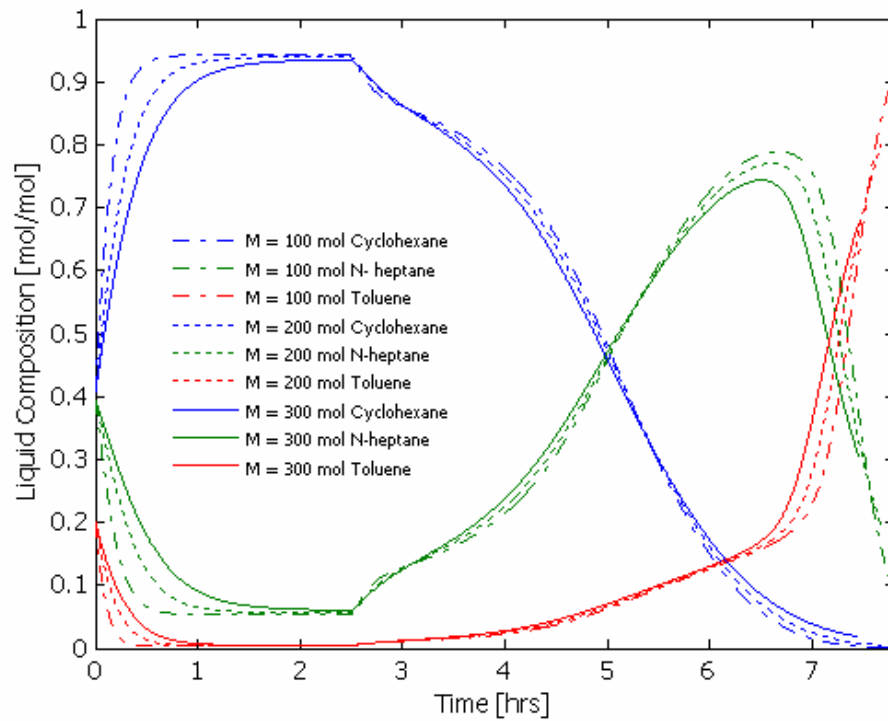


**Figure 7.8** Effect of Condenser Holdup on the Distillate Compositions with One Hour Startup Period.

**Table 7.9** Distillate Compositions at Different Condenser Holdups for 4.5 Hours Elapsed Time.

| Time [hrs] | Condenser Holdup [mol] | Liquid Compositions, mole fractions |            |         |
|------------|------------------------|-------------------------------------|------------|---------|
|            |                        | Cyclohexane                         | N- heptane | Toluene |
| 2.50       | 100                    | 0.9427                              | 0.0535     | 0.0038  |
| 2.50       | 200                    | 0.9393                              | 0.0566     | 0.0041  |
| 2.50       | 300                    | 0.9351                              | 0.0604     | 0.0044  |
| 4.50       | 100                    | 0.6530                              | 0.3097     | 0.0373  |
| 4.50       | 200                    | 0.6359                              | 0.3233     | 0.0408  |
| 4.50       | 300                    | 0.6155                              | 0.3398     | 0.0447  |



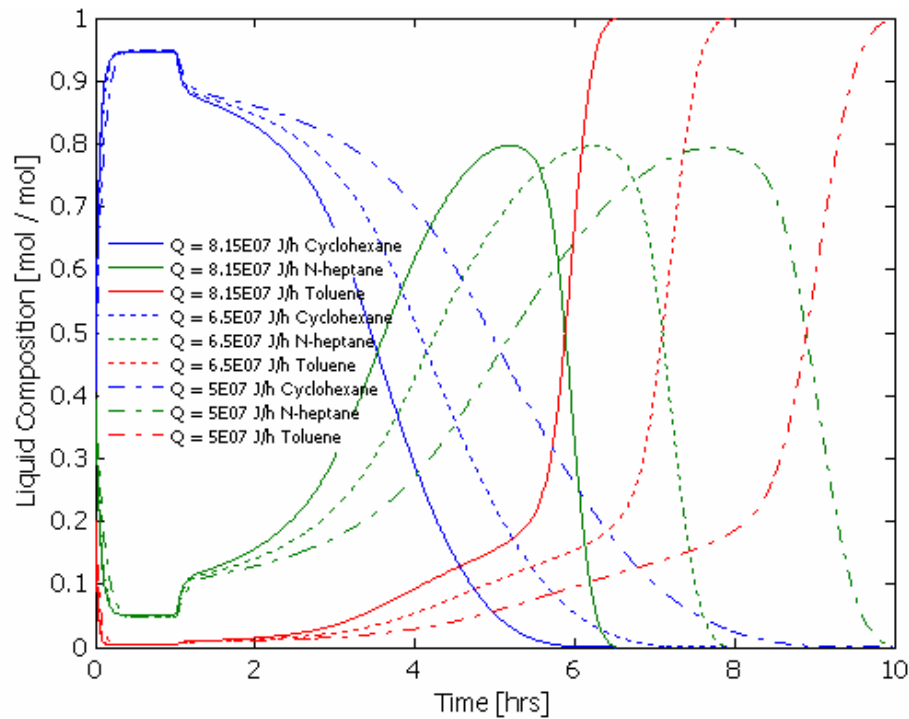


**Figure 7.9** Effect of Condenser Holdup on Distillate Compositions with 2.5 Hours Startup Period.

**Table 7.10** Percentage Change of Distillate Compositions at Different Condenser Holdups.

| Compound    | Condenser Holdups [mol] | Percentage Change (%) |                |
|-------------|-------------------------|-----------------------|----------------|
|             |                         | Time = 2.5 hrs        | Time = 4.5 hrs |
| Cyclohexane | 100                     | -                     | -              |
|             | 200                     | -0.36                 | -2.62          |
|             | 300                     | -0.80                 | -5.74          |
| N – heptane | 100                     | -                     | -              |
|             | 200                     | 5.79                  | 4.39           |
|             | 300                     | 12.90                 | 9.72           |
| Toluene     | 100                     | -                     | -              |
|             | 200                     | 7.90                  | 9.38           |
|             | 300                     | 15.79                 | 19.84          |

Lastly, the effect of heat load is analyzed and is given in Figure 7.10. Heat load is changed from  $8.15 \times 10^7$  J/h to  $6.5 \times 10^7$  and  $5 \times 10^7$  J/h by keeping other parameters constant at base run (height of packing = 1.5m, internal reflux ratio = 0.8 and condenser holdup = 20 moles). The highest purity of cyclohexane is again obtained at startup period. Obtained steady state distillate compositions of the cyclohexane, n-heptane and toluene for simulation run 8 and 9 alter 0.26%, 0.50%, 4.29%, 8.19%, 8.33% and 13.89% of distillate compositions of simulation run 1 at total reflux, respectively, as given in Table 7.11 and 7.12. It is found that the greatest change occurs at the compositions of toluene when the column operates at total reflux. The change of liquid compositions of cyclohexane for simulation run 8 and 9 is nearly equal to the composition of cyclohexane at base run during startup period. This slight change increases as time elapse at production period. The percent increase of the composition of cyclohexane at simulation run 8 and 9 are 2.17% and 3.80% at the end of two hour distillation operation, respectively. 0.7975, 0.7964 and 0.7949 are the maximum purities of n-heptane with the heat loads descending order.



**Figure 7.10** Effect of Heat Load on Distillate Compositions.

The maximum purities of the n-heptane do not change very much as the heat load is changed during the production period. However, the operation time is affected by the heat

load given to the reboiler. It is observed that, as the heat load decreases, operation time for separation increases as expected. Decrease in heat load causes collection of the products in longer time periods. Operation times are 6.54, 7.968 and 9.9963 hrs for the heat loads in descending order, respectively.

**Table 7.11** Distillate Compositions at Different Heat Loads for Two Hours Elapsed Time.

| Time [hrs] | Heat Load [J / h]  | Liquid Compositions, mole fractions |            |         |
|------------|--------------------|-------------------------------------|------------|---------|
|            |                    | Cyclohexane                         | N- heptane | Toluene |
| 1.00       | $8.15 \times 10^7$ | 0.9451                              | 0.0513     | 0.0036  |
| 1.00       | $6.5 \times 10^7$  | 0.9476                              | 0.0491     | 0.0033  |
| 1.00       | $5 \times 10^7$    | 0.9498                              | 0.0471     | 0.0031  |
| 2.00       | $8.15 \times 10^7$ | 0.8285                              | 0.1567     | 0.0148  |
| 2.00       | $6.5 \times 10^7$  | 0.8465                              | 0.1410     | 0.0125  |
| 2.00       | $5 \times 10^7$    | 0.8600                              | 0.1292     | 0.0108  |

**Table 7.12** Percentage Change of Distillate Compositions at Different Heat Loads.

| Compound    | Heat Loads [J/h]   | Percentage Change (%) |             |
|-------------|--------------------|-----------------------|-------------|
|             |                    | Time = 1 hr           | Time = 2 hr |
| Cyclohexane | $8.15 \times 10^7$ | -                     | -           |
|             | $6.5 \times 10^7$  | 0.26                  | 2.17        |
|             | $5 \times 10^7$    | 0.50                  | 3.80        |
| N – heptane | $8.15 \times 10^7$ | -                     | -           |
|             | $6.5 \times 10^7$  | -4.29                 | -10.02      |
|             | $5 \times 10^7$    | -8.19                 | -17.55      |
| Toluene     | $8.15 \times 10^7$ | -                     | -           |
|             | $6.5 \times 10^7$  | -8.33                 | -15.54      |
|             | $5 \times 10^7$    | -13.89                | -27.03      |

Considering the effect of four parameters on the operations of batch packed distillation system, the expected effects are observed which verifies qualitatively simulation code prepared.

## 7.4 Experimental Verification of Simulation Code for Polar Mixtures

The experiments are performed for a binary mixture of ethanol – water as explained in Chapter 6 with total reflux ratio for 5 hrs and then with an external reflux ratio of 0.5 for 1 hr. The experimental data collected for the change of distillate compositions with time are given in Table 7.13. Verification of simulation code for polar mixtures is achieved quantitatively by comparing the results of simulation with the experimental findings.

**Table 7.13** Experimentally Measured Liquid Ethanol Compositions in Mole Fractions in the Distillate.

| <b>Time</b><br><b>[min]</b> | <b><math>x_{\text{EtOH}}</math></b><br><b>[mol / mol]</b> | <b>Time</b><br><b>[min]</b> | <b><math>x_{\text{EtOH}}</math></b><br><b>[mol / mol]</b> |
|-----------------------------|---|-----------------------------|---|
| 0                           | 0.6127  | 180                         | 0.7244  |
| 5                           | 0.7177  | 195                         | 0.7224  |
| 10                          | 0.7283  | 210                         | 0.7231  |
| 15                          | 0.7322  | 225                         | 0.7296  |
| 25                          | 0.7319  | 240                         | 0.7361  |
| 35                          | 0.7393  | 255                         | 0.7370  |
| 40                          | 0.7375  | 270                         | 0.7383  |
| 45                          | 0.7344  | 285                         | 0.7414  |
| 60                          | 0.7240  | 300                         | 0.7476  |
| 75                          | 0.7256  | 305                         | 0.7465  |
| 90                          | 0.7326  | 315                         | 0.7339  |
| 110                         | 0.7422  | 320                         | 0.7190  |
| 120                         | 0.7478  | 325                         | 0.7137  |
| 135                         | 0.7359  | 330                         | 0.7095  |
| 150                         | 0.7301  | 335                         | 0.7034  |
| 165                         | 0.7269  | 340                         | 0.6787  |

Experimental findings are compared with the simulation results using two different predictions for the vapor pressure of the components in terms of temperature: Antoine and Wagner equations, as given below respectively,

$$\log P_V 10^{-5} = A_1 - A_2 \left( \frac{1}{(T - 273.15) + A_3} \right) \quad (7.1)$$

$$\ln \left( \frac{P_V}{P_c} \right) = (1 - X)^{-1} [W_1 X + W_2 X^{1.5} + W_3 X^3 + W_4 X^6] \quad (7.2)$$

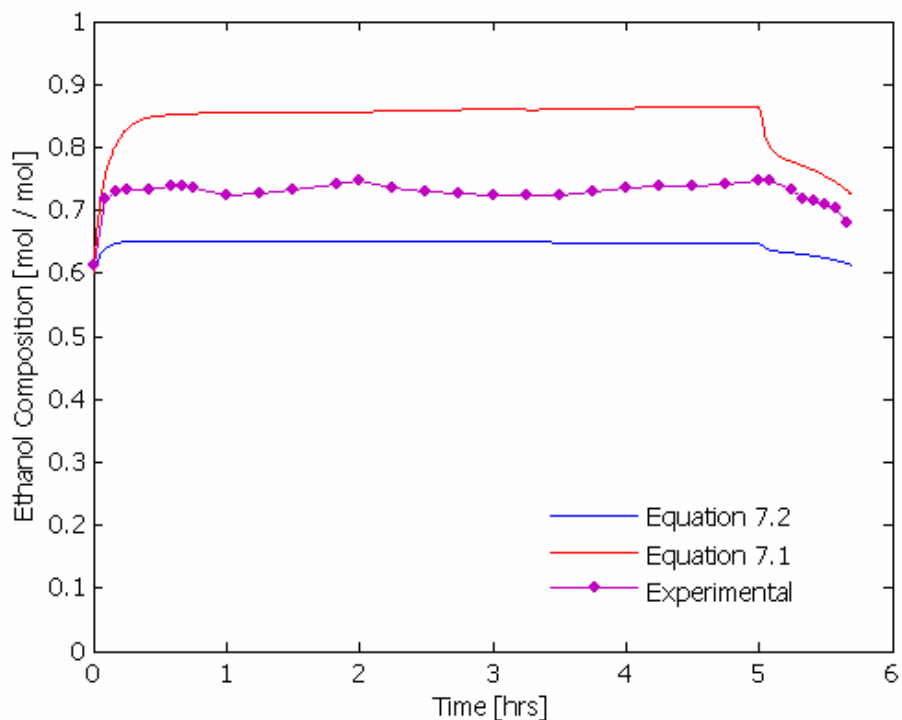
where,

$$X = \left( 1 - \frac{T}{T_c} \right) \quad (7.3)$$

The constants of Equations 7.1 and 7.2 are given in Appendix D (Maier et al. (1998), Reid et al. (1987)).

The comparison of the experimentally obtained dynamic ethanol concentrations of the distillate and their calculated values with simulation are given in Figure 7.11. The experimental data is collected at constant reflux for a longer time in order to make sure that, steady state is attained also experimentally in the time calculated theoretically. However, as can be seen from Figure 7.11, the steady state is reached in 0.583 hr experimentally and in 0.498 hr (Equation 7.1) and 0.252 hr (Equation 7.2) in simulation runs. As expected, the highest purity of liquid ethanol composition is obtained at total reflux both theoretically and experimentally. Reid et al. (1987) stated that, the most accurate results are predicted by using Wagner Equation. Nevertheless, in the evaluation of vapor pressure, the calculated values differ approximately by 3% using Equation 7.1(Antoine) or Equation 7.2 (Wagner). However, when used in the simulation code, the net effect, as can be seen from Figure 7.11, is approximately 15% using Antoine Equation and 12% in using Wagner Equation.

Although distillate compositions obtained experimentally and by simulation differ in the total reflux period, it is observed that, the change of composition of ethanol with time in the production period has similar trends. Thus, experimental verification of the simulation code using polar mixture has been successfully done considering the behavior of the distillate composition with time with an error of 12% in experimental composition values considering Wagner Equation in the simulation for pressure dependence. The reason of this error is thought to be mainly as a result of the physical properties of the used components in the simulation which are found at a lower temperature than the experimental column temperature operated.



**Figure 7.11** Comparison of Results Obtained By Simulation and Experiments.

### 7.5 Determination of Optimum Reflux Ratio

One of the aims of this study is to determine optimum reflux ratio profile to maximize distilled product amount, as stated before. Here, the three component system (cyclohexane, n-heptane and toluene) is considered. CAP, which is a measure developed by Luyben (1988) is used to find the optimum reflux ratio profile.

The number of significant figures of optimized internal reflux ratio is rounded to three while the numbers of significant figures of time to switch to another tank and capacity factor are rounded to two in order to have meaningful values for real applications. The effect of significant figures is analyzed by taking into consideration the calculated values to estimate optimum reflux ratio for the collection of lightest product into product tank 1, P1 which are given in Table 7.14. It is clear from the Table 7.14 that, significant figures of reflux ratio have effect on time to switch to another tank, the amount of molar holdup in tank and capacity factor. Rounding the internal reflux ratio from six to three significant figures results to 0.0867% decrease on time to switch another tank, 0.063% increase on the amount of molar holdup in P1 and  $5.69 \times 10^{-5}\%$  increase on capacity factor whereas rounding the

internal reflux ratio from six to one significant figure results to 18.08% increase on time to switch another tank, 14.18% increase on the amount of molar holdup in P1 and 1.20% decrease on capacity factor. Thus, rounding of the internal reflux ratio to three significant digits is accepted. Moreover, the effect of rounding the significant figures of time to switch another tank is examined for an internal reflux ratio of 0.889. Results of the simulation code for the first product tank with an internal reflux ratio of 0.889 for the times to switch another tank with different significant figures are given at Table 7.15. It is found that as the number of significant figures of time values decreases, the amount of holdup in P1 and capacity factor decrease 0.0283% and 0.0097% with the same internal reflux ratio, respectively. These changes are small. Therefore, time value to switch to another tank is also rounded to two significant figures.

**Table 7.14** Effect of Rounded Value of Internal Reflux Ratio on Capacity Factor.

| <b>Internal Reflux Ratio</b> | <b>Time to Switch Another Tank [hrs]</b> | <b>Molar Holdup in Product Tank 1, P<sub>1</sub> [moles]</b> | <b>Capacity Factor [mol/hr]</b> |
|------------------------------|--|--|---------------------------------|
| 0.889059                     | 3.1134                                   | 634.96   | 175.7233                        |
| 0.889060                     | 3.1134                                   | 634.96   | 175.7218                        |
| 0.889100                     | 3.1155                                   | 635.36   | 175.7310                        |
| <b>0.889000</b>              | <b>3.1107</b>                            | <b>634.48</b>  | <b>175.7234</b>                 |
| 0.890000                     | 3.1581                                   | 642.91   | 175.7500                        |
| 0.900000                     | 3.6762                                   | 725.06   | 173.6175                        |

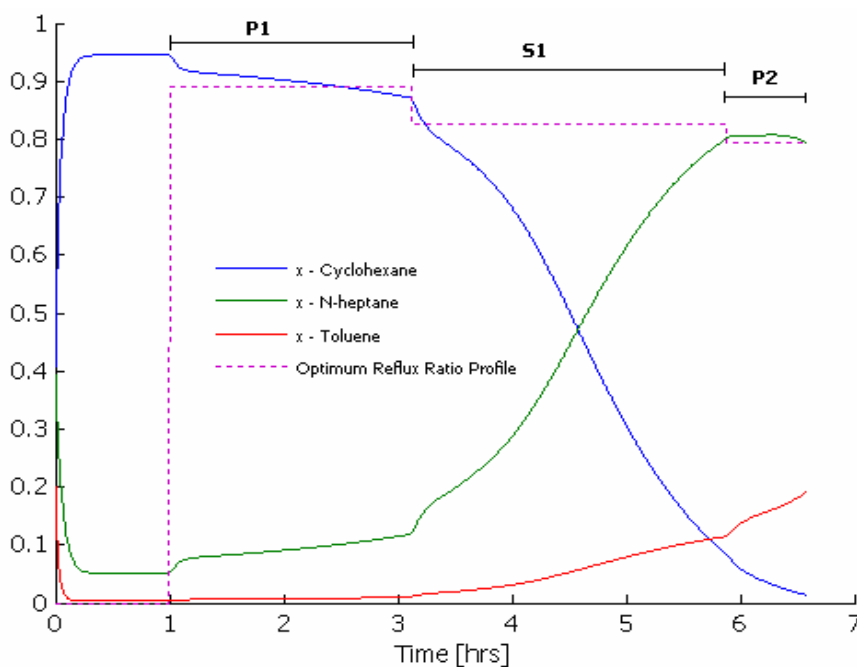
**Table 7.15** List of Some Capacity Factor Values with Rounded Times to Switch another Tank.

| <b>Internal Reflux Ratio</b> | <b>Time to Switch Another Tank [hrs]</b> | <b>Molar Holdup in Product Tank 1, P<sub>1</sub> [moles]</b> | <b>Capacity Factor [mol/hr]</b> |
|------------------------------|--|--|---------------------------------|
| 0.889000                     | 3.1107                                   | 634.48   | 175.7234                        |
| 0.889000                     | 3.1110                                   | 634.57   | 175.7334                        |
| 0.889000                     | 3.1100                                   | 634.30   | 175.7064                        |

The results obtained by simulation utilizing the written optimization code by Bahar (2007) for maximizing product amount is given in Table 7.16 and shown in Figure 7.12.

**Table 7.16** Results of the Optimization Code to Maximize Distilled Product Amount for Cyclohexane – n-Heptane – Toluene Mixture.

| Parameter                  | Optimization Results |             |             |
|----------------------------|----------------------|-------------|-------------|
| Time interval (hr)         | 1 - 3.11             | 3.11 - 5.88 | 5.88 - 6.60 |
| Optimum Reflux Ratio (L/V) | 0.889                | 0.825       | 0.794       |
| CAP (mol/hr)               | 175.71               | 99.42       | 235.07      |



**Figure 7.12** The Liquid Distillate Compositions Obtained under Optimum Reflux Ratio Profile.

During product period, distilled material is collected in storage tanks. Distillate compositions change with time. The lightest product (mostly cyclohexane) is obtained at a higher concentration. Then the second lightest product (mostly n- heptane) is distilled. First and second components are collected into Product 1 ( $P_1$ ) and 2 ( $P_2$ ) tanks, respectively. The "off-spec" material as waste is collected at Slop cut tank 1 ( $S_1$ ). The heaviest product, which



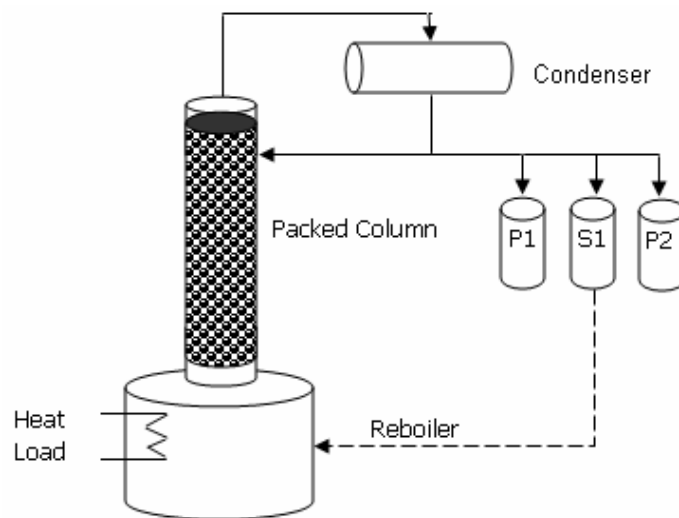
includes toluene mostly, is withdrawn from the reboiler after distillation stops. Thus, at the final stage, the holdups of the packed section and condenser are also drained and collected in the product tank 3 ( $P_3$ ), reboiler.

The desired purities for cyclohexane and second product, n – heptane, are obtained as 0.9 and 0.8, respectively. Thus, the operation is continued for an internal reflux ratio of 0.889 for about 2.11 hours to collect the distillate in the first product tank,  $P_1$ , until distillate composition drops below 0.9. Then, the product, which is “off-spec”, is collected in slop-cut tank  $S_1$  for a certain period of time 2.77 hours with an internal reflux ratio of 0.825 until the second lightest component composition reaches 0.8. Then, the operation continues with an internal reflux ratio of 0.794 until the toluene composition in the reboiler reaches 0.69 for about 0.72 hours. The third compound, toluene is collected in the reboiler. According to simulation results, the purity of the heaviest compound, toluene is 0.69 at reboiler after 6.6 hours operation. However, the purity of toluene decreases to 0.66374 due to the addition of the holdups of the packed section and condenser at the end of the operation. The distillate amounts collected in the tanks are given in Table 7.17.

**Table 7.17** Distillate Compositions and Amounts for the Optimum Reflux Ratio Profile.

| Tank                    | Holdup<br>[moles] | Liquid Compositions, mole fractions |           |         |
|-------------------------|-------------------|-------------------------------------|-----------|---------|
|                         |                   | Cyclohexane                         | N-heptane | Toluene |
| <b>Feed</b>             | 2930.0            | 0.407                               | 0.394     | 0.199   |
| <b><math>P_1</math></b> | 634.3             | 0.8999                              | 0.0926    | 0.0075  |
| <b><math>S_1</math></b> | 1261.0            | 0.4912                              | 0.4521    | 0.0567  |
| <b><math>P_2</math></b> | 369.9             | 0.0393                              | 0.8043    | 0.1564  |
| <b><math>P_3</math></b> | 664.8             | 0.0025                              | 0.3338    | 0.6637  |
| <b>Total</b>            | 2930.0            |                                     |           |         |

Initially, the amount of the mixture fed to the reboiler for separation is 2930 moles. After optimum operation by using reflux ratio profile, 56.96% of the mixture (1669 moles) is separated into the components with desired purities in the first phase of the distillation operation. The molar holdup of slop cut tank,  $S_1$ , is 1261 moles and is large enough not to be wasted. One of the ways to maximize distilled product amount is to recycle holdups of slop cut tank for the next batch distillation operation. Therefore, the contents of the slop cut tank are fed to the reboiler as demonstrated in Figure 7.13 for the proceeding batch.



**Figure 7.13** Schematic View of Recycle Process for the Proceeding Batch Operation.

**Table 7.18** Simulation Results of the First Recycle Process.

| Tank                 | Molar Holdup<br>[mol] | Liquid Compositions, mole fractions |           |         |
|----------------------|-----------------------|-------------------------------------|-----------|---------|
|                      |                       | Cyclohexane                         | N-heptane | Toluene |
| <b>Feed</b>          | 1261.0                | 0.4912                              | 0.4521    | 0.0567  |
| <b>P<sub>1</sub></b> | 456.9                 | 0.9001                              | 0.0977    | 0.0022  |
| <b>S<sub>1</sub></b> | 0                     | 0                                   | 0         | 0       |
| <b>P<sub>2</sub></b> | 0                     | 0                                   | 0         | 0       |
| <b>S<sub>2</sub></b> | 0                     | 0                                   | 0         | 0       |
| <b>P<sub>3</sub></b> | 804.1                 | 0.2570                              | 0.6548    | 0.0882  |
| <b>Total</b>         | 1261.0                |                                     |           |         |

The recycling process is also optimized to get a reflux ratio profile to maximize distilled product amount. It is calculated that the internal reflux ratio of 0.872 must be used for a time interval of 1 – 2.33 hrs after total reflux operation of 1 hour by the simulation code to collect cyclohexane at P<sub>1</sub>. However, for the remaining operation time, the optimum reflux ratio is found to be closer to the total reflux. Therefore, distillation is stopped in 2.33 hr. The results of the recycle process are given in Table 7.18. It is seen that, as a result of the first recycle the gain in the lightest compound is about 457 moles while there is still too much

mixture in P<sub>3</sub>, reboiler. Therefore, one more distillation of the contents of reboiler is done. Results are tabulated in Table 7.19.

**Table 7.19** Simulation Results of Third Distillation Phase.

| Tank                 | Molar Holdup<br>[mol] | Liquid Compositions, mole fractions |           |         |
|----------------------|-----------------------|-------------------------------------|-----------|---------|
|                      |                       | Cyclohexane                         | N-heptane | Toluene |
| <b>Feed</b>          | 804.1                 | 0.2570                              | 0.6548    | 0.0882  |
| <b>P<sub>1</sub></b> | 12.0                  | 0.8993                              | 0.0983    | 0.0024  |
| <b>S<sub>1</sub></b> | 0.0                   | 0.0000                              | 0.0000    | 0.0000  |
| <b>P<sub>2</sub></b> | 0.0                   | 0.0000                              | 0.0000    | 0.0000  |
| <b>S<sub>2</sub></b> | 0.0                   | 0.0000                              | 0.0000    | 0.0000  |
| <b>P<sub>3</sub></b> | 792.1                 | 0.2230                              | 0.6839    | 0.0931  |
| <b>Total</b>         | 804.1                 |                                     |           |         |

In this process, also, only one optimum reflux ratio is calculated as in the case of first recycle process. Internal reflux ratio is found to be 0.2 for the time interval of 1 - 1.39 hrs after total reflux operation of one hour. The molar holdup of separated amount after the third distillation is 12 moles. It is obvious that further distillation will not give any improvement with more operation time spent. Therefore, for further distillation of the reboiler contents other methods such as addition of some fresh feed to reboiler or other column configurations and / or other separation processes can be used. Summary of the results for maximizing the distilled product amount are given in Table 7.20.

As a result of the study, CAP for the optimization with recycling is calculated as 188.86 mol/h. In the calculation of CAP, time necessary for emptying and charging the feed to the reboiler is included for two times (0.5 x 2 hrs) because in the 3<sup>rd</sup> operation, reboiler contents are distilled. The total operation time of separation process for fresh feed and further distillations are included. It is observed that CAP which is 235.07 mol/hr without recycling process decreases to 188.86 mol/hr. This is because; the time necessary for separation of fresh feed and slop cut tanks increases beside the molar holdups of the product tanks. Moreover, the overall separation efficiency is calculated as 72.96% for the fresh feed with recycle whereas it is 56.96% without recycle. Thus, there is a 28% increase in separation efficiency with recycling. Therefore, it can be concluded that recycling of the holdups of the slop cut tanks increases the separation efficiency of the process, significantly.

**Table 7.20** Summary of the Operations with Recycling.

| Tank           | Compound                 | Mole Fractions | Amount of Holdups [mol]   |                           |                           | Total  |
|----------------|--------------------------|----------------|---------------------------|---------------------------|---------------------------|--------|
|                |                          |                | 1 <sup>st</sup> Operation | 2 <sup>nd</sup> Operation | 3 <sup>rd</sup> Operation |        |
| P <sub>1</sub> | Cyclohexane              | 0.90           | 634.3                     | 456.9                     | 12.0                      | 1103.2 |
| P <sub>2</sub> | N- heptane               | 0.80           | 369.9                     | 0.0                       | 0.0                       | 369.9  |
| P <sub>3</sub> | Toluene                  | 0.66           | 664.8                     | 0.0                       | 0.0                       | 664.8  |
| S <sub>1</sub> | "Off – spec"<br>material |                | 1261.0                    | 804.1                     | 792.1                     | 792.1  |
| Total          |                          |                |                           |                           |                           | 2930.0 |

## CHAPTER 8

### CONCLUSIONS AND RECOMMENDATIONS

The objectives of this study were to develop a dynamic model for the simulation of a multicomponent batch packed distillation column with random packing material for mixtures at low operating pressures and to determine optimum reflux ratio profile of the multicomponent batch packed distillation in order to maximize the amount of distillate at a specified concentration for a given time. Thus,

- a dynamic model of a multicomponent batch packed distillation column with random packing material is developed,
- the simulation program is written which can estimate the vapor and liquid compositions, change of temperatures, holdups, liquid and flow rates during the startup and production periods of the distillation operation,
- the accuracy of simulation code is tested qualitatively for hydrocarbon mixtures using a similar study from literature and it is found that, the behaviors of distillate compositions are as expected,
- the accuracy of simulation code is verified experimentally for polar mixtures and it is found that, the experimental findings are in good agreement with the simulation results,
- it is found that the effect of the used vapor pressure – temperature relationship has an important effect on simulation results and Wagner equation gives more accurate results,

- the optimum reflux ratio profile is found to maximize the amount of products. It is found that, optimal reflux ratio profile is a piecewise constant function of time,
- in recycling of holdups of slop cuts for the proceeding batch operation to increase the amount of products, it is found that, one recycling, to collect the lightest compound more effectively, is enough with an increase efficiency of separation about 28%. Further recycle process do not give any further improvements on maximizing the amount of distillate with the specified desired purity of compounds,

As future work,

- written simulation program can be modified for packed distillation column with structured packing materials by changing only the mass transfer and effective interfacial area correlations,
- written simulation program can be modified to be used for reactive packed distillation system column by adding only reaction section,
- any control algorithm can be added to the simulation algorithm easily,
- optimization of reflux ratio to maximize the amount of distillate of a specified concentration for a given time can be improved further to maximize also the profit and to minimize the operation time,
- to increase the efficiency of the distillation operation other than the recycling of holdups of slop cut tank, fresh feed addition to the molar holdup of the slop cut tanks, or other column configurations and /or other separation processes may be utilized.

## REFERENCES

- Alkaya, D., "Determination of A Suitable Measurement Structure for Better Control of Distillation Columns", M. Sc. Thesis, Middle East Technical University, Ankara (1990).
- Attarakih, M., Fara, A. D. and Sayed, S., "Dynamic Modeling of a Packed-Bed Glycerol – Water Distillation Column", *Ind. Eng. Chem. Res.*, Volume 40, pp. 4857 – 4865 (2001).
- Bahar, A., "Control and Simulation Studies for a Multicomponent Reactive Batch Distillation Column", PhD. Thesis, Middle East Technical University, Ankara (2007).
- Betlem, B.H.L., "Batch Distillation Column Low – Order Models for Quality Program Control", *Chemical Engineering Science*, Volume 55, pp. 3187 – 2194 (2000).
- Billet, R., Schultes, M., "Modeling of Pressure Drop in Packed Columns", *Chem. Eng. Technol.*, Volume 14, pp. 89 – 95 (1991).
- Billet, R. and Schultes, M., "Prediction of Mass Transfer Columns with Dumped and Arranged Packings", *Chemical Engineering Research and Design*, Volume 77, Part A, pp.498 – 504, September (1999).
- Bonny, L., "Multicomponent Batch Distillations: Study of Operating Parameters", *Ind. Eng. Chem. Res.*, Volume 38, pp. 4759 – 4768 (1999).
- Bravo, J. L., Patwardhan, A. A., Edgar, T. F., "Influence of Effective Interfacial Areas in the Operation and Control of Packed Distillation Columns", *Ind. Eng. Chem. Res.*, Volume 31, pp. 604 – 608 (1992).
- Choe, Y. S., Luyben, W. L., "Rigorous Dynamic Models of Distillation Columns", *Ind. Eng. Chem. Res.*, Volume 26, pp. 2158 – 2161 (1987).
- Converse, A. O., Gross, G. D., "Optimal Distillate – Rate Policy in Batch Distillation", *Ind. Eng. Chem. Res.*, Volume 2, No. 3, pp. 217 - 221 (1963).

- Coward, I., "The Time – Optimal Problem in Binary Batch Distillation", *Chemical Engineering Science*, Volume 22, pp. 503 – 516 (1967).
- Diwekar, U.M., "Batch Distillation: Simulation, Optimal Design and Control", Taylor & Francis, Washington (1996).
- Djebbar, Y., Narbaitz, R. M., "Neural Network Prediction of Air Stripping  $K_{La}$ ", *Journal of Environmental Engineering*, pp. 451 – 460 May (2002).
- Dokucu, M.T., "Multivariable Model Predictive Controller Design for an Industrial Distillation Column", M. Sc. Thesis, Middle East Technical University, Ankara (2002).
- Fieg, G., Wozny, G., Jeromin, L. and Kruse, C., "Comparison of the Dynamic Behaviour of Packed and Plate Distillation Columns", *Chem. Eng. Technology*, Volume 17, pp. 301 – 306 (1994) .
- Geankoplis, J. C., "Transport Processes and Separation Process Principles (Includes Unit Operations)":Fourth Edition, Prentice Hall (2003).
- Gorak, A., Vogelpoh, A., "Experimental Study of Ternary Distillation in a Packed Column", *Separation Science and Technology*, Volume 20(1), pp.33–61 (1985).
- Gorak, A., "Comments on "Heat, Mass and Momentum Transfer in Packed Bed Distillation Columns" A. Karlström, C. Breitholz, M. Molander *Chem. Eng. Technol.* 15 (1992) No. 1, pp. 1 – 10", *Chem. Eng. Technol.*, Volume 15, pp. 361 – 362 (1992).
- Hanley, B., "The Influence of Flow Maldistribution on The Performance of Columns Containing Random Packings: A Model Study for Constant Relative Volatility and Total Reflux", *Separation and Purification Technology*, Volume 16, pp. 7 – 23 (1999).
- Higler, A., Krishna, R. and Taylor, R., "Nonequilibrium Cell Model for Packed Distillation Columns – The Influence of Maldistribution", *Ind. Eng. Chem. Res.*, Volume 38, pp. 3988 – 3999 (1999).
- Hitch, D.M. and Rousseau, R. W., "Simulation of Continuous – Contact Separation Processes: Multicomponent Batch Distillation", *Ind. Eng. Chem. Res.*, Volume 27, pp. 1466 – 1473 (1988).
- Jahromi, M. F., Domenech, S., Guiglion C., "Modelization of a Packed Batch Distillation Column 1. Establishment of Models", *Chemical Engineering Journal*, Volume 25, No 2, pp. 125 – 135 (1982).



- Jimenez, L., Basualdo, M.S., Gomez, J.C., Toselli, L., Rosa, M., "Nonlinear Dynamic Modeling of Multicomponent Batch Distillation: A Case Study", *Brazilian Journal of Chemical Engineering*, Volume 19, No. 03, pp. 307 – 317, July – September (2002).
- Karacan, S., Cabbar Y., Albaz, M., Hapoğlu H., "The Steady-State And Dynamic Analysis of Packed Distillation Column Based on Partial Differential Approach", *Chemical Engineering and Processing*, Volume 37, pp. 379 – 388 (1998).
- Karlström, A., Breitholtz, C. and Molander, M., "Heat, Mass and Momentum Transfer in Packed Bed Distillation Columns", *Chem. Eng. Technology*, Volume 15, pp. 1 – 10 (1992).
- Kaye, G.W.C., "Tables of Physical and Chemical Constants, and Some Mathematical Functions", 13th ed., McGraw-Hill (Ford) (1969).
- Kim, Y. H., "Optimal Design and Operation of a Multi-product Batch Distillation Column Using Dynamic Model", *Chemical Engineering and Processing*, Volume 38, pp. 61 – 72 (1999).
- Kister, H.Z., Braun, C.F., "Distillation Operation", Inc. New York: Mc Graw Hill, Inc. (1989).
- Klerk, Arno de, "Voidage Variation in Packed Beds at Small Column to Particle Diameter Ratio", *AIChE Journal*, Volume 49, No.8, pp. 2022 – 2029, August (2003).
- Kreul, L. U., Górak, A., Barton, P.I., "Dynamic Rate – Based Model for Multicomponent Batch Distillation", *AIChE Journal*, Volume 45, No.9, pp. 1953 – 1962, September (1999).
- Krishnamurthy, R., and Taylor, R., "Simulation of Packed Distillation and Absorption Columns", *Ind. Eng. Chem. Process Des. Dev.*, Volume 24, pp. 513 – 524 (1985).
- Li, P., Hoo H. P., Wozny G., "Efficient Simulation of Batch Distillation Processes by Using Orthogonal Collocation", *Chem. Eng. Technol.*, Volume 11, pp. 853 – 862 (1998).
- Low, K. H. and Sørensen, E., "Simultaneous Optimal Configuration, Design and Operation of Batch Distillation", *AIChE Journal*, Volume 51, No. 6, pp.1700 – 1713 June (2005).
- Luyben, W. L., "Multicomponent Batch Distillation 1. Ternary Systems with Slop Recycle", *Ind. Eng. Chem. Res.*, Volume 27, pp. 642 – 647 (1988).
- Luyben, W. L., "Process Modeling, Simulation and Control for Chemical Engineers", McGraw-Hill, Singapore, 2nd edition (1989).

- Macias – Salinas, R. and Fair, J. R., "Axial Mixing in Modern Packings, Gas, and Liquid Phases: II. Two Phase Flow", *AIChE Journal*, Volume 46, No.1, pg. 79- 91 (2000).
- Maier, R. W., Brennecke, J. F. and Stadtherr, M. A., "Reliable Computation of Homogeneous Azeotropes", *AIChE Journal*, Volume 44, No. 8, pp. 1745 – 1755, August (1998).
- Miladi, M. M. and Mujtaba, I. M., "Optimisation of Design and Operation Policies of Binary Batch Distillation with Fixed Product Demand", *Computers and Chemical Engineering*, Volume 28, pp. 2377 – 2390, (2004).
- Mori, H., Ito, C., Oda, A. and Aragaki, T., "Total Reflux Simulation of Packed Column Distillation", *Journal of Chemical Engineering of Japan*, Volume 32, No. 1, pp. 69 – 75 (1999).
- Mujtaba, I.M. and Macchietto, S., "Optimal Operation of Multicomponent Batch Distillation – Multiperiod Formulation and Solution", *Computers Chem. Engng.*, Volume 17, No.12, pp.1191 – 1207 (1993).
- Patwardhan, A. A. and Edgar, T. F., "Nonlinear Model Predictive Control of a Packed Distillation Column", *Ind. Eng. Chem. Res.*, Volume 32, pp. 2345 – 2356 (1993).
- Perry, R. H., Green, D. W., Maloney, J. D., "Perry's Chemical Engineers' Handbook", 6th edition, New York: McGraw – Hill (1984).
- Perry, R.H., Green, D.W., Maloney, J.D., "Perry's Chemical Engineer's Handbook", 7<sup>th</sup> edition, McGraw-Hill, New York (1997).
- Piché, S., Iliuta I., Grandjean B. P. A., Larachi F., "A Unified Approach to the Hydraulics and Mass Transfer in Randomly Packed Towers", *Chemical Engineering Science*, Volume 56, pp. 6003 – 6013 (2001a).
- Piché, S., Larachi, F. and Grandjean B. P. A., "Flooding Capacity in Packed Towers: Database, Correlations and Analysis", *Ind. Eng. Chem. Res.*, Volume 40, pp. 476 – 487 (2001b).
- Reid, R.C., Prausnitz, J.M., Poling B. E., "The Properties of Gases and Liquids", 4th Edition, (1987).
- Renon, H. and Prausnitz, J.M., "Local Compositions in Thermodynamic Excess Functions for Liquid Mixtures", *AIChE Journal*, Volume 14, pp. 135 – 144 (1968).

- Repke, J. U., Villain, O., Günter, W., "A Nonequilibrium Model For Three-Phase Distillation in a Packed Column: Modeling and Experiments", *Computers and Chemical Engineering*, Volume 28, pp. 775 – 780 (2004).
- Salimi, F., Depeyre, D., "Comparison between Dynamic Behavior of a Batch Packed and Plate Column", *Computers Chem. Eng.*, Volume 22 (3), pp. 343 – 349 (1998).
- Sandler, S. I., "Chemical and Engineering Thermodynamics", 3rd Edition, John Wiley & Sons, Inc. (1999).
- Skogestad, S. and Morari, M., "Understanding the Dynamic Behavior of Distillation Column", *Ind. Eng. Chem. Res.*, Volume 27, pp. 1848 – 1862 (1988).
- Stichlmair, J., Bravo J.L., Fair, J.R., "General Model for the Prediction of Pressure Drop and Capacity of Countercurrent Gas / Liquid Packed Columns", *Gas Separation and Purification*, Volume 3 (1), pp. 19 – 28, (1989).
- Taiwo, E. A. and Fasesan, S. O., "Model for Dynamic Liquid Hold-up in a Packed Distillation Column", *Ind. Eng. Chem. Res.*, Volume 43, pp. 197 – 202, (2004).
- Tester, J. W., Modell, M., "Thermodynamics and Its Applications", 3rd edition, Upper Saddle River, N. J. : Prentice Hall PTR (1997).
- Tommasi, G. and Rice, P., "Dynamics of Packed Tower Distillation", *Ind. Eng. Chem. Process Des. Develop.*, Volume 9, No. 2, pp. 234 – 243 (1970).
- Yıldız, U., "Multicomponent Batch Distillation Column Simulation and State Observer Design", M. Sc. Thesis, Middle East Technical University, Ankara (2002).
- Wagner, I., Stichlmair, J. and Fair, J.R., "Mass Transfer in Beds of Modern, High – Efficiency Random Packings", *Ind. Eng. Chem. Res.*, Volume 36, pp. 227 – 237 (1997).
- Wajge, R.M., Wilson, J. M., Pekny, J. F. and Reklaitis, G.V., "Investigation of Numerical Solution Approaches to Multicomponent Batch Distillation in Packed Beds", *Ind. Eng. Chem. Res.*, Volume 36, pp. 1738 – 1746 (1997).
- Wang, G. Q., Yuan, X.G., Yu, K. T., "Review of Mass Transfer Correlations for Packed Columns", *Ind. Eng. Chem. Res.*, Volume 44, pp. 8715 – 8729 (2005).
- Wen, X., Shu, Y., Nandakumar, K., and Chuang, K.T., "Predicting Liquid Flow Profile in Randomly Packed Beds from Computer Simulation", *AIChE Journal*, Volume 47, No. 8, pp. 1770 – 1779, August (2001).

## APPENDIX A

### MASS TRANSFER AND EFFECTIVE INTERFACIAL AREA CORRELATIONS FOR RANDOM PACKING MATERIALS

**Table A.1** Correlations for the Gas And / Or Liquid Side Mass Transfer Coefficients for Random Packings\*.

| <b>Author</b>                       | <b>Correlations</b>   |       |
|-------------------------------------|---|-------|
| Sherwood<br>and<br>Holloway         | $k_L = m \left( \frac{D_L}{a_e} \right) \left( \frac{\rho_L u_L}{\mu_L} \right)^{1-n} Sc_L^{0.5}$                               | (A.1) |
| Van<br>Krevelen<br>and<br>Hoftijzer | $k_G = 0.2 \left( \frac{D_G}{d_c} \right) \left( \frac{\rho_L u_L}{a_p \mu_L} \right) 0.8 Sc_G^{1/3}$                           | (A.2) |
|                                     | $k_L = 0.015 \frac{DL}{\left[ \mu_L^2 / (\rho_L^2 g) \right]^{1/3}} \left( \frac{\rho_L u_L}{a_e \mu_L} \right)^{2/3} Sc^{1/3}$ | (A.3) |
| Shulman<br>and de<br>Grouff         | $k_G = 0.0137 (\rho_G u_G)^{0.65} Sc_G^{-2/3}$  | (A.4) |
|                                     | $k_L = m \left( \frac{D_L}{a_e} \right) \left( \frac{\rho_L u_L}{\mu_L} \right)^{1-n} Sc_L^{0.5}$                               | (A.5) |
| Shulman et<br>al.                   | $k_G = 1.195 u_G \left[ \frac{d_p \rho_G u_G}{\mu_G (1 - \varepsilon)} \right]^{-0.36} Sc_G^{-2/3}$                             | (A.6) |
|                                     | $k_L = 25.1 \frac{D_L}{d_p} \left( \frac{d_p \rho_L u_L}{\mu_L} \right)^{0.45} Sc_L^{0.5}$                                      | (A.7) |

---

\* Nomenclature related to the correlations can be found at referenced material, Wang et.al, 2005.

**Table A.1** Correlations for the Gas And / Or Liquid Side Mass Transfer Coefficients for Random Packings\* (Cont'd).

| Author                                 | Correlations  |
|--|---|
| Onda et al.Bravo and Fair              | $k_G = c \left( \frac{D_G}{a_p d_p^2} \right) \left( \frac{\rho_G u_G}{a_p u_G} \right)^{0.7} Sc_G^{1/3} \quad (A.8)$   |
|  | $k_L = \frac{0.0051}{(a_p d_p)^{-0.4}} \left( \frac{\mu_L g}{\rho_L} \right)^{1/3} \left( \frac{\rho_L u_L}{a_e u_L} \right)^{2/3} Sc_L^{-0.5} \quad (A.9)$   |
| Zech and Mersmann                      | $k_G = K_G \frac{D_G}{d_p} \frac{\varepsilon + 0.12}{\varepsilon(1-\varepsilon)^{-1}} \left[ \frac{\rho_G u_G d_p}{(1-\varepsilon)\mu_G} \right]^{2/3} Sc_G^{1/3} \quad (A.10)$                         |
|  | $k_L = K_L \sqrt{\frac{6D_L}{\pi d_{pe}}} \left( \frac{\rho_L g d_{pe}^2}{\sigma_L} \right)^{-0.15} \left( \frac{u_L g d_{pe}}{3} \right)^{1/6} \quad (A.11)$   |
|  | partial wetting:  |
|  | $k_L = \frac{(3.22 \times 10^3) Sc_L^{0.5} D_L}{a_e (1 - \cos \theta)} \left( \frac{\rho_L u_L}{\mu_L} \right)^\alpha \left( \frac{\rho_L^2 g h_{pe}^3}{\mu_L^2} \right)^{0.27} \quad (A.12)$           |
| Mangers and Ponter                     | where $\alpha = 0.49 \left[ 1 - \cos \theta \right]^{0.6} (\rho_L \sigma_L^3 / \mu_L^4 g)^{0.2} \quad (A.13)$   |
|  | complete wetting:   |
|  | $k_L = \frac{2.03 D_L Sc_L^{0.5}}{a_e} \left( \frac{\rho_L u_L}{\mu_L} \right)^{1.44} \left( \frac{\rho_L^2 g h_{pe}^3}{\mu_L^2} \right)^{-0.183} \quad (A.14)$   |
| Shi and Mersmann, Mersmann and Deixler | $k_G = K_G \frac{D_G}{d_{pe}} \left( \frac{\rho_G u_G d_{pe}}{\mu_G} \right)^{2/3} Sc_G^{1/3} \quad (A.15)$   |
|  | $k_L = 0.86 \sqrt{\frac{6D_L}{\pi d_{pe}}} \sqrt{\left\{ \frac{u_L^{1.2} g^{1.3} \sigma_L^{0.3} \varepsilon^{1.2} (1 - 0.93 \cos \theta)^2}{\nu_L^{1.4} \rho_L^{0.3} a_p^{2.4}} \right\}} \quad (A.16)$ |
| Billet and Schultes                    | $k_G = C_G \frac{a_p^{0.5} D_G}{\sqrt{d_h(\varepsilon - h_L)}} \left( \frac{\rho_G u_G}{a_p u_G} \right)^{3/4} Sc_G^{1/3} \quad (A.17)$   |
|  | $k_L = C_L \left( \frac{\rho_L g}{\mu_L} \right)^{1/6} \left( \frac{D_L}{d_h} \right)^{0.5} \left( \frac{u_L}{a_p} \right)^{1/3} \quad (A.18)$  |
| Wagner et al.                          | $k_G = \frac{Z_t^{-0.5}}{C_{pk}} \sqrt{\frac{4\Phi_G D_G u_G}{\pi(\varepsilon - h_t)}} \quad (A.19)$  |
|  | $k_L = \frac{Z_t^{-0.5}}{C_{pk}} \sqrt{\frac{4\Phi_L D_L u_L}{\pi h_t}} \quad (A.20)$   |

\* Nomenclature related to the correlations can be found at referenced material, Wang et.al, 2005.

**Table A.2** Correlations of Effective Interfacial Area for Random Packing Materials\*.

| Authors               | Correlations  |
|-----------------------|---|
| Shulman et al.        | $a_e = m(808\rho_G^{0.5}u_G)^n L^p \frac{BL^{0.57}\mu_L^T}{\rho_L^{0.84}(AL^q - 1)} \left(\frac{\sigma_L}{0.073}\right)^{s-0.262\log L}$ (A.21)   |
| Onda et al.           | $\frac{a_e}{a_p} = 1 - \exp\left[-1.45\left(\frac{\sigma_c}{\sigma_L}\right)^{0.75} \text{Re}_L^{0.1} Fr_L^{-0.05} We_L^{0.2}\right]$ (A.22)  |
| Puranik and Vogelpohl | $\frac{a_w}{a_p} = \left(\frac{a_e}{a_p}\right)_{EV} = \left(\frac{a_e}{a_p}\right)_{CR} = 1.045\left(\frac{\rho_L u_L}{\mu_L a_p}\right)^{0.041} \left(\frac{\rho_L u_L^2}{\sigma_L a_p}\right)^{0.133} \left(\frac{\sigma_c}{\sigma_L}\right)^{0.182}$ (A.23) |
|                       | $\frac{a_{stat}}{a_p} = 0.229 - 0.091 \ln\left(\frac{\rho_L g}{a_p^2 \sigma_L}\right)$ (A.24)   |
|                       | $\frac{a_{dyn}}{a_p} = \left(\frac{a_e}{a_p}\right)_{PA} = \frac{a_w}{a_p} - \frac{a_{stat}}{a_p}$ (A.25)   |
| Kolev                 | $\frac{a_e}{a_p} = 0.583\left(\frac{\nu_L g}{a_p^2 \sigma_L}\right)^{0.49} \left(\frac{u_L^2 a_p}{g}\right)^{0.196} (a_p d_p)^{0.42}$ (A.26)  |
| Zech and Mersmann     | $a_e = K_L \left(\frac{\rho_L u_L}{\mu_L d_{pe}}\right)^{0.5} \left(\frac{\rho_L g d_{pe}^2}{\sigma_L}\right)^{0.45}$ (A.27)  |
| Bravo and Fair        | $\frac{a_e}{a_p} = 0.498 \left(\frac{u_L \mu_L}{\sigma_L} \frac{6\rho_G u_G}{a_p u_G}\right)^{0.392} \frac{\sigma_L^{0.5}}{Z_t^{0.4}}$ (A.28)   |
| Rizzuti et al.        | $a_e = 39\nu_L^{0.7} u_L^{0.326}$ (A.29)  |
| Rizzuti et al.        | $a_e = (2.944 \times 10^4) \nu_L^{0.28} u_L^{0.313} \quad (\nu < 1.54 \times 10^{-6})$ (A.30)   |
|                       | $a_e = 0.165\nu_L^{-0.625} u_L^{0.313} \quad (\nu > 1.54 \times 10^{-6})$ (A.31)  |
| Linek et al.          | $\frac{a_e}{a_p} = 0.0277 \left(\frac{1 + \cos \theta}{2}\right)^{3.477} (a_p d_p)^{1.585} \left(\frac{\rho_L u_L}{a_p u_L}\right)^{0.641 - 0.407(1 + \cos \theta)/2}$ (A.32)   |
| Costa Novella et al.  | $\frac{a_e}{a_p} = a \left(\frac{\rho_L u_L}{a_p u_L}\right)^{0.13} \left(\frac{u_L^2 a_p}{\rho_L \sigma_L}\right)^{0.38} \frac{(\sigma_c / \sigma_L)^{0.18}}{(a_p Z_t)^b}$ (A.33)  |

\* Nomenclature related to the correlations can be found at referenced material, Wang et.al, 2005.

## APPENDIX B

### CONSTANTS OF PRESSURE DROP EQUATION FOR DUMPED PACKING MATERIALS

**Table B.1** Constants for Some Dumped Packing Materials.

| Packing (Ceramic) | Type / size | a [m <sup>2</sup> m <sup>-3</sup> ] | € [-] | C <sub>1</sub> | C <sub>2</sub> | C <sub>3</sub> |
|-------------------|-------------|-------------------------------------|-------|----------------|----------------|----------------|
| Raschig Rings     | 10          | 472                                 | 0.655 | 48             | 8              | 2.0            |
|                   | 10          | 327                                 | 0.657 | 10             | 8              | 1.8            |
|                   | 15          | 314                                 | 0.676 | 48             | 10             | 2.3            |
|                   | 15          | 264                                 | 0.698 | 48             | 8              | 2.0            |
|                   | 30          | 137                                 | 0.775 | 48             | 8              | 2.0            |
|                   | 35          | 126                                 | 0.773 | 48             | 8              | 2.15           |
| Pall Rings        | 25          | 192                                 | 0.742 | 10             | 3              | 1.2            |
|                   | 25          | 219                                 | 0.74  | 1              | 4              | 1.0            |
|                   | 35          | 139                                 | 0.773 | 33             | 7              | 1.4            |
|                   | 35          | 165                                 | 0.76  | 1              | 6              | 1.1            |
| Reflux Rings      | 50          | 120                                 | 0.78  | 75             | 15             | 1.6            |
| Hiflow Rings      | 20          | 291                                 | 0.75  | 10             | 5              | 1.1            |
| Berl Saddles      | 15          | 300                                 | 0.561 | 32             | 6              | 0.9            |
|                   | 35          | 133                                 | 0.75  | 33             | 14             | 1.0            |
| Intalox Saddles   | 20          | 300                                 | 0.672 | 30             | 6              | 1.4            |
|                   | 25          | 183                                 | 0.732 | 32             | 7              | 1.0            |
|                   | 35          | 135                                 | 0.76  | 30             | 6              | 1.2            |

# APPENDIX C

## SIMULATION CODE

### C.1 Main Program Codes

#### C.1.1 Glob\_Decls.m

```
=====
%      Date      By      Explanation
%-----
%      2002      Yıldız      Original Code
%      2006      Ceylan      Modification to Packed Dist.

% #####
% ----- Programming Definitions -----
% #####

% =====
% Simulation Parameters
% =====
% Dummy variables
global Dummy1; global Dummy2; global Dummy3; global Dummy4;
% Output Warnings
global OUT_WARNING;
% tolerance for the decision to make the component fraction zero
global zero_tolerance;
##### Output File ID
%
% Liquid Profile File
global FID_lprofile;
% Vapor Profile File
global FID_vprofile;
% Temperature Profile File
global FID_tprofile;
% Holdup Profile File
global FID_holdup;
% Liquid and Vapor Flowrate Profile File
global FID_lvflow;
% Controller Profile File
global FID_control;
% Tank Outputs Profile File
global FID_tank;
% Optimization Profile File
global FID_opt;

% =====
% ----- End Programming Definitions -----
% #####

% #####
% ----- Real Plant Simulation Parameters -----
% #####

% =====
% Physical System Definitions
% =====
##### Column Specs.
% Number of Components - in the order of volatilities :
%      1st is most volatile and last is least volatile
global NC;
% Number of packed section (increment number)
global NT;
```



```

% Tower cross-sectional area [m2]
global A;
% Specific packing surface area per unit volume of packing volume [m-1]
global ap;
% Packing specific constant (Onda's correlation)
global c;
% Gravitational constant [m.s-2]
global g;
% Packing height [m]
global z_total;
% Bed void fraction (porosity) [m3m-3]
global e_porosity;
% Particle diameter [m]
global dp;
% Constants for predicting pressure drop for column(given at
% J.Stichlmair et al.(1989))[Dimensionless]
global C1;
global C2;
global C3;
% Initial Time
global tstart;
% Measurement Time Step
global DeltaT;
% Displaying Time Step
global disp_DeltaT;
% Time Span of Simulation
global tfinal;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Material Specifications
% Molecular Weight of Components - in the order of volatilities:
% 1st is most volatile and last is least volatile
global Mol_Weight;
% Critical Pressure [Pa]
global Pc;
% Critical Temperature [K]
global Tc;
% Boiling Temperature [K]
global Tb;
% Critical Volume [m3 /kmol]
global Vc;
% Ideal Gas Constant [Pa m3 / (mol.K)]
global R_constant;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Feed Specs.
% Total amount of feed charged to the still pot (moles)
global M_Feed;
% Feed compositions (moles/moles)
global X_Feed;
% =====
% ----- %
% ----- End Real Plant Simulation Parameters ----- %
% #####

```

## C.1.2 Glob\_Initial.m

```

%=====
%      Date      By      Explanation
%=====
%      2002      Yildiz      Original Code
%      2006      Ceylan      Modification to Packed Dist.
% #####
% ----- Programming Initialization ----- %
% ----- %
% =====
% Simulation Parameters Settings
% =====
% Dummy variables
Dummy1=0.0; Dummy2=0.0; Dummy3=0.0; Dummy4=0.0;
% Output Warnings
OUT_WARNING = 1;
% tolerance for the decision to make the component fraction zero
zero_tolerance = 9e-180;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Output File ID Creation
fclose all; Current_Directory = cd;
% Liquid Profile file
delete([Current_Directory '\ ' 'lprofile.txt']);
FID_lprofile = fopen('lprofile.txt','at');
% Vapor Profile file
delete([Current_Directory '\ ' 'vprofile.txt']);
FID_vprofile = fopen('vprofile.txt','at');
% Temperature Profile file
delete([Current_Directory '\ ' 'tprofile.txt']);
FID_tprofile = fopen('tprofile.txt','at');
% Holdup Profile file
delete([Current_Directory '\ ' 'holdup.txt']);
FID_holdup = fopen('holdup.txt','at');
% Liquid and Vapor Flowrate Profile file
delete([Current_Directory '\ ' 'lvflow.txt']);
FID_lvflow = fopen('lvflow.txt','at');
% Controller Profile file
delete([Current_Directory '\ ' 'control.txt']);
FID_control = fopen('control.txt','at');

```

```

% Tank Outputs Profile file
delete([Current_Directory '\ ' 'tank.txt']);
FID_tank = fopen('tank.txt','at');
% Optimization Profile file
% delete([Current_Directory '\ ' 'opt.txt']);
FID_opt = fopen('opt.txt','at');

% =====
% ----- End Programming Initialization -----
% #####

% #####
% ----- Real Plant Simulation Initialization -----
% -----
% =====
% Physical System Settings
% =====
% Column Specs.
% Number of Components - in the order of volatilities :
% 1st is most volatile and last is least volatile
NC = 3;
% Packing height [m]
z_total = 1.5;
% Number of packed section
NT = 40;
% Tower crosssectional area [m2]
D = 0.35;
A = D^2*pi/4;
% Specific packing surface area per unit volume of packing volume [m-1]
ap = 472;
% Packing specific constant (Onda's correlation)
c = 2;
% Gravitational constant [m.s-2]
g = 9.81;
% Bed void fraction (porosity) [m3m-3]
e_porosity = 0.655;
% Particle diameter [m]
dp = 0.010;
% Constants for predicting pressure drop for column(given at
% J.Stichlmair et al.(1989))[Dimensionless]
C1 = 48;
C2 = 8;
C3 = 2;
% Initial Time
tstart = 0.0;
% Measurement Time Step
DeltaT = 0.0003;
% Displaying Time Step
disp_DeltaT = 0.006;
% Time Span of Simulation
tfinal = 8;

% ===== Material Specifications
% Molecular Weight of Components [g/moles] -in the order of volatilities:
% 1st is most volatile and last is least volatile
Mol_Weight = [84.161 100.204 92.141];
% Critical Pressure [Pa]
Pc = [4.1000e006 2.7200e006 4.1000e006];
% Critical Temperature [K]
Tc = [553.58 540.2 591.8];
% Boiling Temperature [K]
Tb = [353.85 371.55 383.75];
% Critical Volume [m3/kmol]
Vc = [0.308 0.428 0.314];
% Ideal Gas Constant [Pa m3 /(mol.K)]
R_constant = 8.314;

% ===== Feed Specs.
% Total amount of feed charged to the still pot (moles)
M_Feed = 2930.0; %1083.034472;%2930.0;
% Feed compositions (moles/moles)[cyclohexane; n-heptane;toluene]
X_Feed = [0.407; 0.394; 0.199];

% =====
% -----End Real Plant Simulation Initialization -----
% #####

```

### C.1.3 Mass\_Hydrocarbons.m

```

% =====
% Date By Explanation
% =====
% 2006 Ceylan Original Code

function [ae, K_G, Mu_g_mixture, Mu_l_mixture]= Mass_Hydrocarbons(i,...
X_frac, Y_frac, Temp, Ro_l_Density, Mw_MolWeight, L_flow, V_flow, ...
Press, Ro_g_Density)

Glob_Decls;
% Since the calculations and predictions of some properties is necessary

```

```

% only for packed section, k value is taken from 1 to NT.

% Number of Packed Sections
k = zeros(NT,1);

k =i-1;
%-----Calculation of Gas(vapor)Viscosity for Pure
% Hydrocarbons at Low Pressures from PERRY's Handbook at Page 2- 363.

% Constant in Equation 2 - 97 in Perry's Handbook (1997)
N = zeros(1);
% Tr = Reduced Temperature [K]
Tr = zeros(1,NC);

% Vapor Viscosity of Compounds [cp] [1 cp = 0.001 kg / (m.s)]
Mu_g_compound = zeros(NT,NC);
for i = 1:NC,
    Tr(i) = Temp(k+1) / Tc(i);
    if Tr <= 1.5
        N = 0.0003400 * Tr(i)^0.94;
    else
        N = 0.0001778*(4.58*Tr(i) - 1.67)^0.625;
    end
    Mu_g_compound(k,i) = (4.6000e-004* N * Mol_Weight(i)^0.5*Pc(i)^(2/3))/...
        (Tc(i)^(1/6));
end

%-----Prediction of Vapor Viscosity of Gaseous Hydrocarbons
% and Mixtures of Hydrocarbons at Low Pressures.

% Vapor Viscosity of Gaseous Mixture [cp]
Mu_g_mixture = zeros(NT,1);

sum = 0;
Mu_g_mixture(k) = 0;
for i=1: NC,
    for j=1:NC,
        if j~=i
            QW1 = (Mu_g_compound(k,i)/Mu_g_compound(k,j))^0.5;
            QW2 = (Mol_Weight(j)/Mol_Weight(i))^0.25;
            QW3 = (sqrt(8)*(1+(Mol_Weight(i)/Mol_Weight(j))^0.5));
            QW = (1 + (QW1*QW2)^2) / QW3;
        else
            QW = 0;
        end
        sum = QW*(X_frac(k+1,j)/X_frac(k+1,i)) + sum;
    end
    Mu_g_mixture(k) = Mu_g_compound(k,i)/(1 + sum) + Mu_g_mixture(k);
end
Mu_g_mixture = Mu_g_mixture * 0.001; % Conversion of cp to [ kg/(m.s)]

% -----Calculation of Liquid Viscosity of a Component
% Equation(2-112)(Perry's handbook) for Pure Hydrocarbon and Pure
% Nonhydrocarbon Liquids

% Liquid Viscosity of Compound [kg/(m.s)]
Mu_l_compound = zeros(NT,NC);
% Adjusted Carbon Number
Nstar = zeros(1,NC);
% Constant Determined by Equations(2-114 or 2-115) in Perry's Handbook
To = zeros(1,NC);
% Constant Determined by Equations(2-117 or 2-118) in Perry's Handbook
Ba = zeros(1,NC);
% Constant Determined by Equations(2-116) in Perry's Handbook
B = zeros(1,NC);
% Group Contribution Number Determined from Table 2- 398 in Perry's Handbook
del_B = zeros(1,NC);

Nstar = [7.48 7 7.60];
del_B = [-85.5433 0 -34.6356];
for i=1:NC,
    if Nstar(i) <= 20
        To(i) = 28.86 + 37.439 * Nstar(i) - 1.3547 * Nstar(i)^2 + ...
            0.02076 * Nstar(i)^3;
        Ba(i) = 24.79 + 66.885 * Nstar(i) - 1.3173 * Nstar(i)^2 - ...
            0.00377 * Nstar(i)^3;
    else
        To(i) = 8.164 * Nstar(i) + 238.59;
        Ba(i) = 530.59 + 13.740 * Nstar(i);
    end
    B(i) = Ba(i) + del_B(i);
    Mu_l_compound(k,i) = (10^(B(i)*((1/Temp(k+1))-(1/To(i)))) * 0.001;
end
% Kendall and Monroe's Equation is used for determining
% the liquid viscosity of defined hydrocarbon mixtures. (Equation 2-119) in
% PERRY's Handbook.

% Liquid Viscosity of Mixture [kg/(m.s)]
Mu_l_mixture = zeros(NT,1);

for i = 1:NC,
    Mu_l_mixture(k) = Mu_l_compound(k,i)^(1/3)*X_frac(k+1,i) + Mu_l_mixture(k);
end;
Mu_l_mixture(k) = (Mu_l_mixture(k))^3;

%-----Calculation of Liquid Surface Tension of a Component
% by Using Equation 2 - 166 and 2- 167 in Perry's Handbook

```

```

% Liquid Surface Tension of a Component
Sigma_l_stensionco = zeros(NT,NC);
% Constant in Equation (2- 167) in Perry's Handbook
QS = zeros(NT,NC);
% Reduced Boiling Temperature [K]
Tbr = zeros(1,NC);

for i=1:NC,
    Tbr(i) = Tb(i) / Tc(i);
    QS(k,i) = 0.1207 * (1+ (Tbr(i)* (log(Pc(i))-11.5261))/(1-Tbr(i))) - 0.281;
    Sigma_l_stensionco(k,i) = (4.6010e-004) * (Pc(i)^(2/3)) * (Tc(i)^(1/3)) * ...
        QS(k,i) * (1-Tbr(i))^(11/9);
end;

%-----Calculation of Liquid Surface Tension of a Mixture

% Liquid Surface Tension of the Mixture [mN/m]
Sigma_l_surfacetension = zeros(NT,1);
% as initial point
P1 = 0;

for i=1:NC,
    P1 = X_frac(k+1,i) / ((1e-3*Ro_l_Density(k+1))/Mw_MolWeight(k+1)) + P1;
end
P = 1 / P1;
F0 = 0;
for i = 1:NC,
    for j=1:NC,
        F1 = P^2*(X_frac(k+1,i)/(1e-3*Ro_l_Density(k+1)/Mw_MolWeight(k+1)))...
            *(X_frac(k+1,j)/(1e-3*Ro_l_Density(k+1)/Mw_MolWeight(k+1)))...
            *(Sigma_l_stensionco(k,i)*(Sigma_l_stensionco(k,j)))^0.5;
        F0 = F1 + F0;
    end
end
Sigma_l_surfacetension(k) = F0; % [mN /m]
%-----
% Effective Interfacial Area
%-----Calculation of Reynolds Number
% Liquid Phase Reynolds Number [Dimensionless]
Reynolds_l = zeros(NT,1);
% Gas Phase Reynolds Number [Dimensionless]
Reynolds_g = zeros(NT,1);

Reynolds_l(k) = (L_flow(2)*Mw_MolWeight(k+1))/(A*ap*Mu_l_mixture(k)*3600);
Reynolds_g(k) = (V_flow(2)*Mw_MolWeight(k+1)) / (A*ap*Mu_g_mixture(k)*3600);

%-----Calculation of Froude Number
% Liquid Phase Froude Number [Dimensionless]
Froude_l = zeros(NT,1);

Froude_l(k) = (L_flow(2)^2 * ap*Mw_MolWeight(k+1)^2)/(Ro_l_Density(k+1)^2*A^2* g*3600*3600);

%-----Calculation of Weber Number
% Liquid Phase Weber Number [Dimensionless]
Weber_l = zeros(NT,1);

Weber_l(k) = ((L_flow(2)^2*Mw_MolWeight(k+1)^2)/(ap * Ro_l_Density(k+1))*...
    Sigma_l_surfacetension(k)*A^2*g))*(1/((3600^2)*0.0001019716213));
% Critical Surface Tension of Packing Material [mN/m]
Sigma_l_criticalsurfacetension = zeros(1);
Sigma_l_criticalsurfacetension = 73;

% Effective Interfacial Area [1/m]
ae = zeros(k,1);
%-----Calculation of effective interfacial area
ae(k) = ap * (1- exp(-1*1.45*...
    ((Sigma_l_criticalsurfacetension/Sigma_l_surfacetension(k))^0.75)* ...
    (Reynolds_l(k)^0.1)*(Froude_l(k)^(-1*0.05))*(Weber_l(k)^0.2)));

%-----
% Mass Transfer Coefficient
%-----
%-----Calculation of Gas Diffusion Coefficient
% Molar Volume [m3/kmol]
V = zeros(1,NC);
% Gas Diffusion Coefficient of the Mixture [m2 /sec]
D_g_mixture = zeros(NT,1);

sum = 0;
for i = 1:NC,
    for j=1:NC,
        if i == j
            else
                V(i) = 0.285*(Vc(i)*1000)^1.048;
                V(j) = 0.285*(Vc(j)*1000)^1.048;
                D = (0.1014*((Temp(k+1)*(9/5))^1.5))*(((1/Mol_Weight(i))+...
                    (1/Mol_Weight(j)))^0.5)/(Press(k+1)*(0.0001450377377)*...
                    (((V(i)^(1/3))+(V(j)^(1/3))^2)));

                % Unit Conversion from ft2/hr to m2/sec
                D = D*(0.09290304)/3600;
                sum = Y_frac(k+1,j)* D^-1 +sum;
            end
        end;
    end;
end;
D_g_mixture(k) = ((1 - Y_frac(k+1,i))/(sum));

```

```

%-----Calculation of Gas Phase Schmidt Number
% Gas Phase Schmidt Number [Dimensionless]
Schmidt_g(k) = Mu_g_mixture(k) / (Ro_g_Density(k+1) * D_g_mixture(k));

%-----Calculation of Packing Efficiency Number(Er)
% Packing Efficiency Number [Dimensionless]
Er = zeros(1);

Er = 1 / (ap * dp);
%-----
% Gas Phase Mass Transfer Coefficient [mol/h.m2]
K_G = zeros(NT,1);

% Constant of Onda Mass Transfer Correlation
c = zeros(1);

if dp < (15* 10^-3)
    c = 2;
else
    c = 5.23;
end;
%-----Calculation of Mass Transfer Coefficient
K_G(k) = c*3600* D_g_mixture(k)*ap*(Reynolds_g(k))^0.7 * ...
Schmidt_g(k) ^ (1/3) * (Er)^2*[Press(k+1)/(R_constant*Temp(k+1))];

%-----End of Simulation -----
%-----
%=====

```

## C.1.4 Mass\_Polar.m

```

%=====
%      Date      By      Explanation
%=====
%      2006      Ceylan      Original Code
%=====

function [ae, K_G, Mu_g_mixture, Mu_l_mixture]= Mass_Polar(L_flow, ...
    V_flow, X_frac, Mw_MolWeight, Ro_l_Density, Ro_g_Density, Temp,...
    Press, Y_frac)

Glob_Decls;
% Since the calculations and predictions of some properties is necessary
% only for packed section, k value is taken from 1 to NT.

% Number of Packed Sections
k = zeros(NT,1);
% Vapor Viscosity of Compounds [kg / (m.s)] [1 cp = 0.001 kg / (m.s)]
Mu_g_compound = zeros(NT,NC);
% Vapor Viscosity of Gaseous Mixture [cp]
Mu_g_mixture = zeros(NT,1);
% Liquid Viscosity of Compound [kg/(m.s)]
Mu_l_compound = zeros(NT,NC);
% Liquid Viscosity of Mixture [kg/(m.s)]
Mu_l_mixture = zeros(NT,1);
% Liquid Surface Tension of a Component [mN/m]
Sigma_l_stensionco = zeros(NT,NC);
% Liquid Surface Tension of the Mixture [mN/m]
Sigma_mixture = zeros(NT,1);
% Critical Surface Tension of Packing Material [mN/m]
Sigma_l_criticalsurfacetension = zeros(1);
% Liquid Phase Reynolds Number [Dimensionless]
Reynolds_l = zeros(NT,1);
% Gas Phase Reynolds Number [Dimensionless]
Reynolds_g = zeros(NT,1);
% Liquid Phase Froude Number [Dimensionless]
Froude_l = zeros(NT,1);
% Liquid Phase Weber Number [Dimensionless]
Weber_l = zeros(NT,1);
% Effective Interfacial Area [1/m]
ae = zeros(NT,1);
% Gas Diffusion Coefficient of the Mixture [m2 /sec]
D_g_mixture = zeros(NT,1);
% Gas Phase Schmidt Number [Dimensionless]
Schmidt_g = zeros(NT,1);
% Packing Efficiency Number [Dimensionless]
Er = zeros(1);
% Gas Phase Mass Transfer Coefficient [mol/h.m2]
K_G = zeros(NT,1);
% Constant of Onda Mass Transfer Correlation
c = zeros(1);

if dp < (15* 10^-3)
    c = 2;
else
    c = 5.23;
end;
sum = 0;
% k =i-1;
for k = 1:NT,
    Mu_g_compound(k,1) = 8.6088*10^-3 ;% T = 20 C
    Mu_g_compound(k,2) = 125 * 10^-4 ; %T = 20C
    for i=1:NC,

```

```

for j=1:NC,
    if j~=i
        QW1= (Mu_g_compound(k,i)/Mu_g_compound(k,j))^0.5;
        QW2=(Mol_Weight(j)/Mol_Weight(i))^0.25;
        QW3 = (sqrt(8)*1+(Mol_Weight(i)/Mol_Weight(j)))^0.5);
        QW = (1 + (QW1*QW2)^2) / QW3;
    else
        QW = 0;
    end
    sum = QW*(X_frac(k+1,j)/X_frac(k+1,i)) + sum;
end
Mu_g_mixture(k)= Mu_g_compound(k,i)/(1 + sum)+ Mu_g_mixture(k);
end
Mu_g_mixture = Mu_g_mixture * 0.001; % Conversion of cp to [ kg/(m.s)]

Mu_l_compound(k,1) = 1.7840*10^-3;% T = 20 C
Mu_l_compound(k,2) = 1*10^-3 ;% T = 20 C

sum = 0;
for i=1:NC,
    sum = X_frac(k+1,i)*log(Mu_l_compound(k,i))+ sum;
end
Mu_l_mixture(k,1) = exp(sum);

Sigma_l_criticalsurfacetension = 73;
Sigma_l_stensionco(k,1)= 22.39 ;% T = 20 C
Sigma_l_stensionco(k,2)= 72.75 ;% T = 20 C

% Calculation of Densities for Ethanol and Water [g/ml]
Density_Ethanol = zeros(1);
Density_Water = zeros(1);
Density_Ethanol = 0.0015;
Density_Water = 0.001;

% Calculation of surface tension of mixture[m3/kmol]
Molar_Volume = zeros(1,NC);
Molar_Volume(1,1) = 1 / (Density_Ethanol*10^3/Mol_Weight(1,1));
Molar_Volume(1,2) = 1 / (Density_Water*10^3 / Mol_Weight(1,2));

Constant_mixture = zeros(NC,NC);

Constant_mixture(k,2)=log10([(X_frac(k+1,2)*...
    Molar_Volume(2))/(X_frac(k+1,1)*Molar_Volume(1))]*[X_frac(k+1,2)*...
    Molar_Volume(2)+X_frac(k+1,1)*Molar_Volume(1)]^-1)+...
    +(44.1*2/Temp(k+1))*[Sigma_l_stensionco(k,1)*Molar_Volume(1)^(2/3)...
    /2 - Sigma_l_stensionco(k,2)*Molar_Volume(2)^(2/3)];
Constant_mixture(k,2) = Constant_mixture(k,2)/(1+Constant_mixture(k,2));
Constant_mixture(k,1) = 1 - Constant_mixture(k,2);
Sigma_mixture(k,1)= Constant_mixture(k,1)*...
    Sigma_l_stensionco(k,1)^(1/4) +Constant_mixture(k,2)*...
    Sigma_l_stensionco(k,2)^(1/4);

D_g_mixture(k,1) = 1.3927*10^-5;

%-----Calculation of Dimensionless Numbers
Reynolds_l(k)=(L_flow(2)*Mw_MolWeight(k+1))/(A*ap*Mu_l_mixture(k)*3600);
Reynolds_g(k)=(V_flow(2)*Mw_MolWeight(k+1))/(A*ap*Mu_g_mixture(k)*3600);
Froude_l(k) = (L_flow(2)^2 * ap*Mw_MolWeight(k+1)^2)/...
    (Ro_l_Density(k+1)^2*A^2* g*3600*3600);
Weber_l(k) = ((L_flow(2)^2*Mw_MolWeight(k+1)^2)/(ap * Ro_l_Density(k+1)...
    * Sigma_mixture(k)*A^2*g))*(1/((3600^2)*0.0001019716213));
Schmidt_g(k) = Mu_g_mixture(k) / (Ro_g_Density(k+1) * D_g_mixture(k));
Er = 1 / (ap * dp);
%-----
% Effective Interfacial Area & Mass Transfer Coefficient
%-----
ae(k) = ap * (1- exp(-1*1.45*...
    ((Sigma_l_criticalsurfacetension/Sigma_mixture(k))^0.75)*...
    (Reynolds_l(k)^0.1)*(Froude_l(k)^(-1*0.05))*(Weber_l(k)^0.2));
K_G(k) = c*3600* D_g_mixture(k)*ap*(Reynolds_g(k))^0.7 *...
    Schmidt_g(k)^(1/3) * (Er)^2*[Press(k+1)/(R_constant*Temp(k+1))];
end;
#####

```

## C.1.5 PressureProfile.m

```

%=====
%      Date          By          Explanation
%=====
%      2002          Yıldız      Original Code
%      2006          Ceylan      Modification to Packed Dist.
%=====
% PressureProfile
% Return
%      pressure profile through the column (Pa)
% for a given
%      Still pot and reflux drum pressures (Pa)
%=====
function [fP_Packed]=PressureProfile(fP_Pot,fP_Drum)
Glob_dec;

if (size(fP_Pot)~=1 | size(fP_Drum)~=1)

```

```

        error('fP_Pot and/or fP_Drum are not scalar(s). [PressureProfile]');
    end;
    fP_Packed = zeros(NT,1);

    fdelP_Packed = (fP_Pot-fP_Drum)/NT;

    for i=1:NT; fP_Packed(i) = fP_Pot - i*fdelP_Packed; end;

    % end PressureProfile

```

## C.1.6 Plant\_File\_Packed.m

```

=====
%      Date          By          Explanation
=====
%      2002          Yildiz      Original Code
%      2006          Ceylan      Modification to Packed Dist.
=====

function Plant_File_Packed
% function [CAP] = plant_file_packed(R_first_prod)

% =====
% Clear Command Window
clc;
% Include All Global Variables
Glob_Decls;
% Initialize thermo_LIBRARY.dll
thermo_Init(0);
% Initialize All Global Variables
display('Global variables are initializing ...');
Glob_Initial;
display('Global variables have been initialized. ');
fprintf('\nEnter type of compounds that you want to study in the simulation algorithm.\n\n');
fprintf('Please enter 1 for hydrocarbons , enter 2 for polar mixtures... \n\n');
type_of_compound = input(':: ');

% =====
% Step Time and Time Span
% =====
% Time Span of Simulation
num_step = round((tfinal - tstart)/DeltaT);
% Current Simulation Time
t = tstart;
% Current Displaying Time Step
disp_t = tstart;

% =====
% Differential Packed Section Height
% =====
% Initial Height of Packed Section
zstart = 0.0;
% A Differential Packed Section Height
DeltaZ = zeros(1);
DeltaZ = z_total / NT;
% Current Simulation Height
z = zstart;

% =====
% Initialize Real Plant
% =====
#####Initial Operation Parameters.
% Boiler Load [J/hour]
Q_Boiler = 8.15e7;
% One Over Reflux Ratio (D/L0) [dimensionless]
R_Ratio_inv = 0.0;
% Distillate Flow Rate [mol/hour]
D_DistillRate = 0.0;
% Amount of Product Distilled [moles]
M_Distilled = 0.0;
% Initial Still Pot, Packed Section, Reflux Drum Pressure [Pa]
Press(1,1) = 101000.0;
Press(NT+2,1) = 101600.0;
Press(2:NT+1,1) = PressureProfile(Press(1), Press(NT+2));
fdelP_Packed = 1;
fdelP_Packed1 = 0;
#####Mass Transfer Coefficients
%Effective Interfacial Area [m-1]
ae = zeros(NT,1);
%Mass Transfer Coefficient [mol/(h.m)]
K_G = zeros(NT,1);
% Rate Equation of Mass Transfer [mol/h] (Rate = KG * A * ae)
Rate = zeros(NT,1);
% Equivalent Diameter of Packing [m]
dp_equi = zeros (1);
dp_equi = 6 * (1-e_porosity)/ ap;

% R_ratio_opt = R_first_prod;

#####
% In that part, pressure drop is calculated via trial and error procedure.
% To predict, first trial for pressure drop, initial assumptions are made
% by taking Yildiz (2002) as reference.
#####

```

```

while (fdelP_Packed)- fdelP_Packed1 > 0.001,
fdelP_Packed1 = fdelP_Packed;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Liquid Compositions
%Abbreviation for Still Pot, Packed Section, Reflux Drum =(Sp-Ps-Rd)
% Instantaneous (Sp-Ps-Rd) Liquid Compositions [moles/moles]
X_frac = zeros(NT+2,NC);
% Initially (Sp-Ps-Rd) Liquid Compositions (equal to that of the feed)
for i=1:NT+2;
    X_frac(i,:) = X_Feed.';
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Temperature
% Instantaneous (Sp-Ps-Rd) Temperature [K]
Temp = zeros(NT+2,1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Vapor Compositions
% Instantaneous (Sp-Ps-Rd) Vapor Compositions [moles/moles]
Y_frac = zeros(NT+2,NC);
% Equilibrium Compositions of Vapor
Y_frac_eqm = zeros(NT+2,NC);

% -----Bubble Point Calculation
Dummy1 = 360.0;
for i=1:NT+2;
    if (type_of_compound == 1)
        [Temp(i), Dummy3] = ...
            thermo_Equilibrium_Hydrocarbons(Dummy1, Press(i), X_frac(i,:));
    else
        [Temp(i), Dummy3] = ...
            thermo_Equilibrium_Polar(Dummy1, Press(i), X_frac(i,:));
    end;
    Y_frac_eqm(i,:) = Dummy3;
    Y_frac(i,:) = Y_frac_eqm(i,:);
end;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Physical Properties
% Instantaneous (Sp-Ps-Rd) Liquid Phase Enthalpy [J/moles]
H_l_Enthalpy = zeros(NT+2,1);
% Instantaneous (Sp-Ps-Rd) Vapor Phase Enthalpy [J/moles]
H_v_Enthalpy = zeros(NT+2,1);
% Instantaneous (Sp-Ps-Rd) Liquid Phase Average Density [kg/m3]
Ro_l_Density = zeros(NT+2,1);
% Instantaneous (Sp-Ps-Rd) Vapor Phase Average Density [kg/m3]
Ro_g_Density = zeros(NT+2,1);
% Instantaneous (Sp-Ps-Rd) Average Molecular Weight [kg/mol]
Mw_MolWeight = zeros(NT+2,1);

% -----Calculate Specific Phase Enthalpies
for i=1:NT+2;
    [H_l_Enthalpy(i), H_v_Enthalpy(i)] = ...
        thermo_Enthalpy(Temp(i), Press(i), X_frac(i,:), Y_frac(i,:));
end;

% -----Calculate Liquid Phase Average Density and Average Molecular Weight
for i=1:NT+2;
    [Mw_MolWeight(i), Ro_l_Density(i)] = ...
        thermo_L_Density(Temp(i), Press(i), X_frac(i,:));
end;
for i=1:NT+2;
    [Mw_MolWeight(i), Ro_g_Density(i)] = ...
        thermo_G_Density(Temp(i), Press(i), X_frac(i,:));
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Liquid and Vapor Flowrates
% Instantaneous Liquid Flow Rates Leaving Packed Section and Reflux Drum [moles/hour]
% L_flow(1) : dummy
L_flow = zeros(NT+2,1);
% Instantaneous Vapor Flow Rates Leaving Still Pot and Packed Section [moles/hour]
% V_flow(NT+2) : dummy
V_flow = zeros(NT+2,1);
% -----Calculate Vapor and Liquid Flow Rates
% Calculate Still Pot Vapor Flow Rate
V_flow(1) = Q_Boiler/(H_v_Enthalpy(1) - H_l_Enthalpy(2));

% Initialize Packed Section Vapor Flow Rates
% V_flow(NT+2) : dummy
for i=2:NT+1;
    V_flow(i) = V_flow(1);
end;

% Initialize Packed Section and Reflux Drum Liquid Flow Rates
% L_flow(1) : dummy
for i=2:NT+2;
    L_flow(i) = V_flow(1);
end;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Liquid Holdups
% Instantaneous (Sp-Ps-Rd) Total Holdup Amount [moles]
M_Holdup = zeros(NT+2,1);
% -----Calculate Initial Molar Holdups
% Initial Reflux Drum Liquid Holdup [moles]
M_Holdup(NT+2) = 20.0; %
% Liquid Holdup throughout Packed Section[-]
ho = zeros(1);
%Froude Number [-]
Fr = zeros(1);

% Calculate Packed Section Molar Holdup [mol]
for i=2:NT+1;
    Fr = (((L_flow(2)*Mw_MolWeight(i)) / (Ro_l_Density(i)*A*3600))^2)*...
        (ap/(g*e_porosity^(4.65)));

```



```

        ho = 0.555 * ((Fr)^(1/3));
        M_Holdup(i) = ((A * ho * Ro_l_Density(i)*DeltaZ)/(Mw_MolWeight(i));
end;
% Calculate Still Pot Molar Holdup [mol]
M_Holdup(1) = M_Feed - sum(M_Holdup(2:NT+2));
%Calculation of Liquid and Gas Velocities [m/sec]
% Liquid Velocity [m/sec]
UL = zeros(NT+2,1);
% Gas Velocity [m/sec]
UG = zeros(NT+2,1);
% Design Liquid Velocity [m/sec]
UL_design = zeros(NT+2,1);
% Design Gas Velocity [m/sec]
UG_design = zeros(NT+2,1);
% Liquid Kinematic Viscosity [m2/sec]
v_l = zeros(NT+2,1);
% Gas Kinematic Viscosity [m2/sec]
v_g = zeros(NT+2,1);

for i=2:NT+1,
    UL_design(i) = (L_flow(2)*Mw_MolWeight(i)) / (Ro_l_Density(i)*A*3600);
    UG_design(i) = (V_flow(2)*Mw_MolWeight(i)) / (Ro_g_Density(i)*A*3600);
    if (type_of_compound == 1)
        [ae, K_G, Mu_g_mixture, Mu_l_mixture] = Mass_Hydrocarbons(i,...
            X_frac, Y_frac, Temp, Ro_l_Density, Mw_MolWeight, L_flow,...
            V_flow, Press, Ro_g_Density);
    else
        [ae, K_G, Mu_g_mixture, Mu_l_mixture] = Mass_Polar(L_flow, ...
            V_flow, X_frac, Mw_MolWeight, Ro_l_Density, Ro_g_Density,...
            Temp, Press, Y_frac);
    end;
    v_l(i) = Mu_l_mixture(i-1) / Ro_l_Density(i);
    v_g(i) = Mu_g_mixture(i-1) / Ro_g_Density(i);
end;

UL = UL_design;
UG = UG_design;

% Calculation of Reynolds Number
[Re_G]= P_reynolds(UG, dp_equi, v_g);
% Calculation of Friction Factor for a Single Particle
[fo] = P_friction(C1,C2,C3,Re_G);
% Calculation of Dry Pressure Drop
[del_p_dry ]= P_dry_pressure_drop(fo, e_porosity, Ro_g_Density, z_total, UG, dp_equi);
% Calculation of liquid holdup below the loading point
[ho, FrL]= P_holdup(UL, ap, g, e_porosity);
% Calculation of exponent constant at Equation 10 given by J.Stichlmair et al.(1989)
[c_exp]= P_constant_exp(C1,C2,Re_G,fo);
%Calculation of irrigated pressure drop
[fdelP_Packed]= P_pressure_drop_irrigated(del_p_dry,Ro_l_Density,...
    z_total,ho,e_porosity,c_exp,g);

Press(NT+2,1) = 101600.0;
Press(1,1) = Press(NT+2,1) - fdelP_Packed;
Press(2:NT+1,1) = PressureProfile(Press(1), Press(NT+2));

end
fprintf('\n\nThe irrigated pressure drop is %f (Pa).\n',fdelP_Packed)

%-----Prediction of flooding point by using correlation given by
%J. Stichlmair et al.(1989)
% Abbreviation for Equation (18) given by J. Stichlmair et. al.(1989)
Abbreviation = zeros(1);
Abbreviation = (fdelP_Packed / (Ro_l_Density(2) * g * z_total))^2;

% Rearranged Equation(18) given by J. Stichlmair et. al.(1989)
B_Rearranged = zeros(1);
B_Rearranged = (1 / Abbreviation) - [(40*(2+c_exp)/3)*ho/(1-e_porosity+ho*...
    [1+20*Abbreviation])] - [(186*ho)/(e_porosity-ho*[1+20*Abbreviation])];

% Flooding Point Liquid Velocity [m/sec]
UL_flooding = zeros(NT+2,1);
% Flooding Point Gas Velocity [m/sec]
UG_flooding = zeros(NT+2,1);
% Flooding Pressure [N/m2]
fdelP_Packed_flooding = zeros(1);

UG_flooding = UG_design;% Initial assumption

UG = UG_flooding;
while B_Rearranged > zero_tolerance,
    UG(2) = UG(2) + 0.001;
    fdelP_Packed1 = 1;
    while fdelP_Packed - fdelP_Packed1>0.01
        fdelP_Packed1 = fdelP_Packed;
        % Calculation of Reynolds Number
        [Re_G]= P_reynolds(UG, dp_equi, v_g);
        % Calculation of Friction Factor for a Single Particle
        [fo] = P_friction(C1,C2,C3,Re_G);
        % Calculation of Dry Pressure Drop
        [del_p_dry ]= P_dry_pressure_drop(fo, e_porosity, Ro_g_Density,...
            z_total, UG, dp_equi);
        % Calculation of liquid holdup below the loading point
        [ho, FrL]= P_holdup(UL, ap, g, e_porosity);
        % Calculation of exponent constant at Equation 10 given by J.Stichlmair et al.(1989)
        [c_exp]= P_constant_exp(C1,C2,Re_G,fo);
        %Calculation of irrigated pressure drop
        [fdelP_Packed]= P_pressure_drop_irrigated(del_p_dry,Ro_l_Density,...

```

```

        z_total,ho,e_porosity,c_exp,g);
    end
    Abbreviation = (fdelP_Packed / (Ro_l_Density(2) * g * z_total))^2;
    B_Rearranged = (1 / Abbreviation) - [(40*(2+c_exp)/3)*ho/(1-e_porosity+ho...
        *[1+20*Abbreviation])] - [(186*ho)/(e_porosity-ho*[1+20*Abbreviation])];
end
%-----End of Pressure Drop Prediction -----
fdelP_Packed_flooding = fdelP_Packed;
for i=1:NT+2,
    UG_flooding(i)= UG(2);
end
% Percent of flooding [Dimensionless]
percent = zeros(1);

percent = (UG_design(2) / UG_flooding(2))*100;
fprintf('The flooding pressure drop is %f (Pa).\n',fdelP_Packed_flooding)
fprintf('The design gas rate is %f percent of flooding.\n\n', percent)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%Define and Initialize storage tank compositions
% Instantaneous compositions in storage tanks (moles/moles) (2*(NC-1)xNC)
Tank_X_frac = zeros(2*(NC-1)+1, NC);
% Instantaneous total holdup amount in storage tanks (moles)
Tank_M_Holdup = zeros(2*(NC-1)+1, 1);
% =====
% Initialize Controller
% =====
% CONT_SetPoints      : Controller Set Points of Product Specifications
% CONT_Num_Oper_Stage : Number of different operation stage
% CONT_DistillProfile : Distillate Flow rate values of different operation stages (Products and/or Slopcuts)
[CONT_SetPoints, CONT_Num_Oper_Stage, CONT_DistillProfile] = INIT_CONTROL;
% Define Controller Outputs
CONT_QBoiler = zeros(1,1);
CONT_RRatio_inv = zeros(1,1);
% Initialize Activated Tank Index
Tank_Activated = 0;
% Initialize Current Stage Number (startup:0, )
CONT_Curr_Stage = 0;
% =====
% Integration Starts
% =====
for i=0:num_step
    if t >= disp_t;
        % Write Plant Data to Screen
        write_plant_to_scr(t, X_frac, R_Ratio_inv);

        % Write Plant Data to File
        write_plant_to_file(t, X_frac, Y_frac, Temp, M_Holdup, L_flow, V_flow);

        % Write estimator and controller data to file
        write_estcont_to_file(t, CONT_QBoiler, CONT_RRatio_inv);

        % Write tank compositions and holdups
        write_tank_to_file(t, Tank_X_frac, Tank_M_Holdup);
        %Tank_X_frac
        %D_DistillRate

        disp_t = disp_t + disp_DeltaT;
    end;

    % Real Plant Variables
    % % % % % % % % % % X_frac, Y_frac, Temp, Y_frac_eqm, t,z
    % % % % % % % % % % H_l_Enthalpy, H_v_Enthalpy, Ro_l_Density, Mw_MolWeight, Ro_g_Density
    % % % % % % % % % % M_Holdup, L_flow, V_flow,
    % % % % % % % % % % Q_Boiler, D_DistillRate, R_Ratio_inv, M_Distilled, Press,
    % % % % % % % % % % t_prv, M_Holdup_prv, H_l_Enthalpy_prv, z_prv
    % % % % % % % % % % Del_M_Holdup, Del_H_l_Enthalpy, Del_M_Hl

    % Keep Current Parameters for Future Use
    [t_prv, M_Holdup_prv, H_l_Enthalpy_prv, X_frac_prv, Y_frac_prv] = ...
        P_Keep_Current_Vars(t, M_Holdup, H_l_Enthalpy, X_frac, Y_frac);

    % Find Derivatives
    [DX_frac, DY_frac] = P_f(t, X_frac, Y_frac, Temp, H_l_Enthalpy, ...
        H_v_Enthalpy, Ro_l_Density, Ro_g_Density, Mw_MolWeight, M_Holdup, ...
        L_flow, V_flow, Q_Boiler, D_DistillRate, R_Ratio_inv, Press, ...
        t_prv, M_Holdup_prv, H_l_Enthalpy_prv, Y_frac_eqm, DeltaT, ...
        DeltaZ, X_frac_prv, type_of_compound);

    % Take Integration
    [t, z, X_frac, Y_frac] = P_Int_Euler(t, z, DeltaT, DeltaZ, X_frac, ...
        DX_frac, Y_frac, DY_frac, Y_frac_eqm);

    % Normalizes Plant States
    % Vapor Phase
    [Y_frac] = P_Normalize_States_Y(z, Y_frac);
    % Liquid Phase
    [X_frac] = P_Normalize_States_X(t, X_frac);

    % Run controller (Find new real plant inputs)
    if (t >= 0 & t < 1.0)
        % Run the open-loop system with using predefined switching
        % times and corresponding distillate flowrates (or reflux ratios)
        [CONT_QBoiler, CONT_RRatio_inv, Tank_Activated] = CONTROL(t, ...
            X_frac, Q_Boiler, CONT_SetPoints, CONT_Num_Oper_Stage, CONT_DistillProfile);
    else

```

```

% Run the closed-loop system with the actual composition feed-back
[CONT_QBoiler, CONT_RRatio_inv, Tank_Activated, CONT_Curr_Stage] = CONTROL_real(t,...
    X_frac, Q_Boiler, R_Ratio_inv, CONT_SetPoints, CONT_Num_Oper_Stage,...
    CONT_DistillProfile, Tank_Activated, Tank_X_frac, CONT_Curr_Stage);
% if (CONT_Curr_Stage == 5)
%   if (Tank_Activated==3)
%     Tank_M_Holdup(5,1) = M_Holdup(1,1) + sum(M_Holdup(2:NT+1)) + M_Holdup(NT+2);
%     sum1 = zeros(1,NC);
%     sum2 = zeros(1, NC);
%     for i=2:NT+1;
%       sum2 = M_Holdup(i)*X_frac(i,:);
%       sum1 = sum2 + sum1;
%     end
%     Tank_X_frac(5,:) = [(X_frac(1,:)*M_Holdup(1,1) + sum1 + ...
%       M_Holdup(NT+2)*X_frac(NT+2,:))] / [Tank_M_Holdup(5,1)];
%   end
% end
end

% Manipulate real plant inputs by controller outputs
Q_Boiler = CONT_QBoiler;
R_Ratio_inv = CONT_RRatio_inv;

% Stop simulation when the distillation finishes
if (Tank_Activated == -1)
    break;
end;
% Find New Physical Variables
[Y_frac_eqm, Temp, H_l_Enthalpy, H_v_Enthalpy, Ro_l_Density, Mw_MolWeight] = ...
    P_Calc_New_Phys_Vars(t, X_frac, Press, Temp, Y_frac, type_of_compound);

% Calculate New Packed Section Molar Holdups
[M_Holdup] = P_Calc_New_Mol_Packed_Section_Holdup(t, Ro_l_Density,...
    Mw_MolWeight, M_Holdup, L_flow);

% Calculate New Reflux-Drum Molar Holdup
[M_Holdup] = P_Calc_New_Mol_Drum_Holdup(t, M_Holdup);

% Calculate New Approximated Derivatives
[Del_M_Holdup, Del_H_l_Enthalpy, Del_M_Hl] = P_Calc_New_Apprx_Deriv(t, ...
    M_Holdup, H_l_Enthalpy, t_prv, M_Holdup_prv, H_l_Enthalpy_prv);

% Calculate New Distillate, Vapor and Liquid Flow Rates
if R_Ratio_inv == 0; %% For Total Reflux Condition
    [D_DistillRate, L_flow, V_flow] = P_Calc_LV_for_Total_Reflux(t, ...
        H_l_Enthalpy, H_v_Enthalpy, M_Holdup, Del_M_Holdup, ...
        Del_H_l_Enthalpy, Del_M_Hl, Q_Boiler, R_Ratio_inv);
else %% For Distillate Withdrawal
    [D_DistillRate, L_flow, V_flow] = P_Calc_LV_for_Finite_Reflux(t, ...
        H_l_Enthalpy, H_v_Enthalpy, M_Holdup, Del_M_Holdup, ...
        Del_H_l_Enthalpy, Del_M_Hl, Q_Boiler, R_Ratio_inv);
end;

% Calculate current holdup amount and composition in storage tanks
[Tank_X_frac, Tank_M_Holdup] = P_Calc_Tanks(t, DeltaT, Tank_X_frac, ...
    Tank_M_Holdup, X_frac(NT+2,:), D_DistillRate, Tank_Activated);

% Calculate New Amount of Product Distilled
[M_Distilled] = P_Calc_New_Distilled_Amount(t, t_prv, D_DistillRate, M_Distilled);

% Calculate New Still Pot Holdup
[M_Holdup] = P_Calc_New_Mol_Still_Holdup(t, M_Holdup, M_Distilled);

end;

if (R_ratio_opt == 1)
    CAP = 0;
else
    CAP = (Tank_M_Holdup(1,1) + Tank_M_Holdup(3,1)+ Tank_M_Holdup(5,1)) / (t+0.5);
end;
cap_fact = CAP;

% Write optimization results to file
write_opt_to_file(t,R_ratio_opt,Tank_M_Holdup(1,1),Tank_M_Holdup(2,1),...
    Tank_M_Holdup(3,1),Tank_M_Holdup(4,1),Tank_M_Holdup(5,1),cap_fact);

% =====
% Close Output Files
% =====
% Liquid Profile file
fclose(FID_lprofile);
% Vapor Profile file
fclose(FID_vprofile);
% Temperature Profile file
fclose(FID_tprofile);
% Holdup Profile file
fclose(FID_holdup);
% Liquid and Vapor Flowrate Profile file
fclose(FID_lvflow);
% Controller Outputs file
fclose(FID_control);
% Tank Outputs file
fclose(FID_tank);
% Optimization Outputs file
fclose(FID_opt);

% =====
% Simulation finishes

```

```

% =====
fprintf('Simulation finished successfully.\n\n');

% -----End of Main Program function Plant_File_Packed

% #####
% -----Real Plant Simulation Functions-----
% -----

% Predict Pressure Drop (Dry, Irrigated and Flooding)
% =====
function [Re_G] = P_reynolds(UG, dp_equi, v_g)
% Reynolds Number of Gas Calculated by Equivalent Diameter of Packing
Re_G = zeros(1);
Re_G = (UG(2)*dp_equi)/v_g(2);
%end_P_reynolds

function [fo] = P_friction(C1,C2,C3,Re_G)
% Friction Factor [Dimensionless]
fo = zeros(1);
fo = (C1 / Re_G)+(C2/(Re_G^0.5))+ C3;
%end_P_friction

function [del_p_dry ]= P_dry_pressure_drop(fo, e_porosity, Ro_g_Density, ...
z_total, UG, dp_equi)
% Dry Pressure Drop [N/m2]
del_p_dry = zeros(1);
del_p_dry = (3/4)*fo*((1-e_porosity)/(e_porosity^4.65))* Ro_g_Density(2)*...
z_total * UG(2)^2 / dp_equi;
% end_P_dry_pressure_drop

function [ho, FrL]= P_holdup(UL, ap, g, e_porosity);
% Liquid Froude Number
FrL = zeros(1);
% Liquid Holdup
ho = zeros(1);
FrL = ((UL(2)^2)*ap) / (g * e_porosity^4.65);
ho = 0.555*FrL^(1/3);
%end_P_holdup

function [c_exp]=P_constant_exp(C1,C2,Re_G,fo)
%Exponent c for Calculation of the Irrigated Pressure Drop Constant
%at Equation 10 given by J.Stichlmair et al.(1989)
c_exp = zeros(1);
c_exp = ((-C1/Re_G)+(-C2/(2*Re_G^(1/2))))/fo;
%end_P_constant_exp

function [fdelP_Packed]=P_pressure_drop_irrigated(del_p_dry,Ro_l_Density,...
z_total,ho,e_porosity,c_exp,g)

del_p_irrigatedl = del_p_dry;
A1 = (del_p_irrigatedl/(Ro_l_Density(2)*g*z_total));
A2 = (1 + 20 * (A1^2));
A3 = (ho/e_porosity)*A2;
A4 = (1-e_porosity*(1-A3));
A5 = (A4/(1-e_porosity))^(2+c_exp)/3;
A6 = (1-A3)^(-1*4.65);
fdelP_Packed = del_p_dry*A5*A6;

while del_p_irrigatedl - fdelP_Packed > 0.001;
    fdelP_Packed = del_p_irrigatedl;
    A1 = (del_p_irrigatedl/(Ro_l_Density(2)*g*z_total));
    A2 = (1 + 20 * (A1^2));
    A3 = (ho/e_porosity)*A2;
    A4 = (1-e_porosity*(1-A3));
    A5 = (A4/(1-e_porosity))^(2+c_exp)/3;
    A6 = (1-A3)^(-4.65);
    del_p_irrigatedl = del_p_dry*A5*A6;
end
%end_P_pressure_drop_irrigated

% =====
% P_f
% Return the derivatives for the continuous states.
% =====

function [DX_frac, DY_frac] = P_f(t, X_frac, Y_frac, Temp, H_l_Enthalpy,...
H_v_Enthalpy, Ro_l_Density, Ro_g_Density, Mw_MolWeight,M_Holdup,...
L_flow, V_flow,Q_Boiler, D_DistillRate, R_Ratio_inv, Press, t_prv, M_Holdup_prv,...
H_l_Enthalpy_prv, Y_frac_eqm, DeltaT, DeltaZ,X_frac_prv,type_of_compound)
Glob_Decls;
% Instantaneous (Sp-Ps-Rd)Liquid and Vapor Compositions Derivatives [moles/moles/hour]
DX_frac = zeros(size(X_frac));
DY_frac = zeros(size(Y_frac));

%-----Calculation of Mass Transfer Coefficient
for i =2:NT+1;
    if (type_of_compound == 1)
        [ae, K_G,Mu_g_mixture, Mu_l_mixture] = Mass_Hydrocarbons(i,X_frac,...
Y_frac, Temp, Ro_l_Density, Mw_MolWeight, L_flow, V_flow, Press, Ro_g_Density);
    else
        [ae, K_G, Mu_g_mixture, Mu_l_mixture]= Mass_Polar(L_flow, V_flow,...
X_frac, Mw_MolWeight, Ro_l_Density, Ro_g_Density, Temp, Press, Y_frac);
    end;
    Rate(i-1) = K_G(i-1)*A*ae(i-1);
end;
%-----Calculation of derivatives for compositions

```

```

for j=1:NC;
% Still Pot
DX_frac(1,j) = ( L_flow(2)*(X_frac(2,j)-X_frac(1,j)) - V_flow(1)*...
(Y_frac_eqm(1,j)-X_frac(1,j)) ) / M_Holdup(1);
Y_frac(1,j) = Y_frac_eqm(1,j);

% Packed Section
for i = 2:NT+1,
C = zeros(1);
C = (-Rate(i-1)/V_flow(i));
DY_frac(i,j)= C* (Y_frac(i,j) - Y_frac_eqm(i,j));
end;
% Following equation is a pde written for a packed differential
% section. The approximation to (del_x / del_z) was made by using the
% upward Euler method.{(x(i+1)-x(i))/deltaZ}
for i=2:NT+1,
DX_frac(i,j) = Rate(i-1)*(Y_frac(i,j)-Y_frac_eqm(i,j))/[(M_Holdup(i)/DeltaZ]+ ...
(L_flow(i)/(M_Holdup(i)/DeltaZ))*(X_frac(i+1,j)- X_frac(i,j))/(DeltaZ)];
end;
% Reflux Drum
DX_frac(NT+2,j) = ( V_flow(NT+1)*(Y_frac(NT+1,j)-X_frac(NT+2,j)) ) / M_Holdup(NT+2);
end
% end mdlDerivatives
=====
% P_Int_Euler
% Return
% Take integral
% given
% Time: t;
% Integration Step: delta_t;
% Previous States: X_frac;
% Derivatives: DX_frac;
=====
%In that part, for the prediction of the new liquid and vapor compositions
% upward Euler method is used as an approximation method for ordinary and
% partial differential equations.
=====
function [t_new, z_new, X_frac_new, Y_frac_new] = P_Int_Euler(t,z,DeltaT, ...
DeltaZ, X_frac, DX_frac, Y_frac, DY_frac, Y_frac_eqm)
Glob_Decls;
t_new = t + DeltaT;
z_new = z + DeltaZ;

% New Liquid Compositions
X_frac_new = X_frac+ DX_frac*DeltaT;

% New Vapor Compositions
for j=1:NC,
Y_frac(1,j) = Y_frac_eqm(1,j);
for i = 2:NT+1,
Y_frac(i,j) = Y_frac(i,j)+ DeltaZ* DY_frac(i,j);
Y_frac(i,j) = (Y_frac(i-1,j)+Y_frac(i,j))/2;
end
Y_frac(NT+2,j) = Y_frac(NT+1,j);
end
Y_frac_new = Y_frac;

%end Int_Euler
=====
% P_Keep_Current_Vars
% Return
% Keeps current parameters for future use
% given
% Time: t; Variables at t: ...;
=====
function [t_prv, M_Holdup_prv, H_l_Enthalpy_prv, X_frac_prv, Y_frac_prv] = ...
P_Keep_Current_Vars(t, M_Holdup, H_l_Enthalpy, X_frac, Y_frac);
% Previous step time
t_prv = t;
% Previous step (Sp-Ps-Rd) total holdup amount (moles)
M_Holdup_prv = M_Holdup;
% Previous step (Sp-Ps-Rd) liquid phase enthalpy (J/moles)
H_l_Enthalpy_prv = H_l_Enthalpy;
% Previous step (Sp-Ps-Rd) liquid phase composition (mol / mol)
X_frac_prv = X_frac;
% Previous step (Sp-Ps-Rd) vapor phase composition (mol / mol)
Y_frac_prv = Y_frac;

%end P_Keep_Current_Vars
=====
% P_Normalize_States_Y
% Return
% Normalizes Plant States for Vapor Phase
% given
% Time: t; States: Y_frac_Pr;
=====
function [Y_frac] = P_Normalize_States_Y(t, Y_frac_Pr)
Glob_Decls;

% Set size
Y_frac = Y_frac_Pr;

% *****Make the low compositions zero*****
% Check for (Sp-Ps-Rd) vapor compositions (moles/moles)
for i=1:NT+2;
for j=1:NC;
if isnan(Y_frac_Pr(i,j));

```

```

        error(['Stage no ', num2str(i) , ' component ', num2str(j),...
            ' vapor fraction is Nan' ]);
    end;
    if Y_frac_Pr(i,j) < zero_tolerance;
        % display(['Stage no ',num2str(i),' composition of comp. ',...
            % num2str(j), ' ( ',num2str(Y_frac_Pr(i,j)),') made zero']);
        Y_frac(i,j) = 0.0;
    end;
end;
end;

% *****Normalize the vapor compositions (moles/moles)*****
% Normalize (Sp-Ps-Rd) vapor compositions (moles/moles)
for i=1:NT+2;
    dummy1 = sum(Y_frac_Pr(i,:));
    if ~(dummy1 > 0.0);
        % error(['Sum of comp. vapor fraction in the Stage no ',...
            % num2str(i), ' is zero. ', num2str(Y_frac_Pr(i,:)) ]);
    else
        Y_frac(i,:) = Y_frac_Pr(i,:) / dummy1;
    end;
end;

% *****Check compositions are in the limit of [0,1] *****
% Check (Sp-Ps-Rd) vapor compositions (moles/moles)
for i=1:NT+2;
    for j=1:NC;
        if (Y_frac(i,j)<0 | Y_frac(i,j)>1);
            % error(['Vapor Composition out of limit ! - [Normalize_States] Y_frac(',...
                % num2str(i),',:)= ', num2str(Y_frac(i,:)) ] );
        end;
    end;
end;

%end P_Normalize_States_Y
%=====
% P_Normalize_States
% Return
% Normalizes Plant States for Liquid Phase
% given
% Time: t; States: X_frac_Pr;
%=====
function [X_frac] = P_Normalize_States_X(t, X_frac_Pr)
Glob_Decls;

% Set Size
X_frac = X_frac_Pr;

% *****Make the low compositions zero*****
% Check for (Sp-Ps-Rd) liquid compositions (moles/moles)
for i=1:NT+2;
    for j=1:NC;
        if isnan(X_frac_Pr(i,j));
            error(['Stage no ', num2str(i) , ' component ', num2str(j),...
                ' liquid fraction is Nan' ] );
        end;
        if X_frac_Pr(i,j) < zero_tolerance;
            % display(['Stage no ',num2str(i),' composition of comp. ',...
                % num2str(j), ' ( ',num2str(X_frac_Pr(i,j)),') made zero']);
            X_frac(i,j) = 0.0;
        end;
    end;
end;

% *****Normalize the liquid compositions(moles/moles)*****
% Normalize (Sp-Ps-Rd) liquid compositions (moles/moles)
for i=1:NT+2;
    dummy1 = sum(X_frac_Pr(i,:));
    if ~(dummy1 > 0.0);
        error(['Sum of comp. fraction in the Stage no ',...
            num2str(i), ' is zero. ', num2str(X_frac_Pr(i,:)) ]);
    else
        X_frac(i,:) = X_frac_Pr(i,:) / dummy1;
    end;
end;

% *****Check compositions are in the limit of [0,1]*****
% Check (Sp-Ps-Rd) liquid compositions (moles/moles)
for i=1:NT+2;
    for j=1:NC;
        if (X_frac(i,j)<0 | X_frac(i,j)>1);
            error(['Composition out of limit ! - [Normalize_States] X_frac(',...
                num2str(i),',:)= ', num2str(X_frac(i,:)) ] );
        end;
    end;
end;

%end P_Normalize_States_X
%=====
% P_Calc_New_Phys_Vars
% Return
% Find New Physical Variables
% given
% Time, t; Previous system variables (Temp at previous time step is for initial guess)
%=====
function [Y_frac_eqm, Temp, H_l_Enthalpy, H_v_Enthalpy, Ro_l_Density, Mw_MolWeight] = ...
    P_Calc_New_Phys_Vars(t, X_frac, Press, TempPr, Y_frac, type_of_compound)
Glob_Decls;

```

```

% Set Sizes
Y_frac_eqm = zeros(size(X_frac));
Temp = zeros(size(TempPr));
H_l_Enthalpy = zeros(size(TempPr));
H_v_Enthalpy = zeros(size(TempPr));
Ro_l_Density = zeros(size(TempPr));
Mw_MolWeight = zeros(size(TempPr));

%-----Bubble Point Calculation
% Calculate (Sp-Ps-Rd) Temperature and Vapor Compositions [moles/moles]
for i=1:NT+2;
    if (type_of_compound == 1)
        [Temp(i), Dummy3] = thermo_Equilibrium_Hydrocarbons(TempPr(i), Press(i), X_frac(i,:));
    else
        [Temp(i), Dummy3] = thermo_Equilibrium_Polar(TempPr(i), Press(i), X_frac(i,:));
    end;
    Y_frac_eqm(i,:) = Dummy3;
end;
for j = 1:NC,
    for i=2:NT+1,
        Y_frac_eqm(i,j) = [Y_frac_eqm(i,j) + Y_frac_eqm(i+1,j)]/2;
    end
end

%-----Calculate Specific Phase Enthalpies
% Calculate (Sp-Ps-Rd)Liquid and Vapor Phase Enthalpies [J/moles]
for i=1:NT+2,
    [H_l_Enthalpy(i), H_v_Enthalpy(i)] = thermo_Enthalpy(Temp(i), Press(i), X_frac(i,:), Y_frac(i,:));
end;
%-----Calculate Liquid Phase Average Density and Average Molecular Weight
% Calculate (Sp-Ps-Rd) Liquid and Gas Phase Average Densities [kg/m3]
% Calculate (Sp-Ps-Rd) Average Molecular Weight [kg/mol]
for i=1:NT+2;
    [Mw_MolWeight(i), Ro_l_Density(i)] = thermo_L_Density(Temp(i), Press(i), X_frac(i,:));
    [Mw_MolWeight(i), Ro_g_Density(i)] = thermo_G_Density(Temp(i), Press(i), X_frac(i,:));
end;

%end P_Calc_New_Phys_Vars
%=====
% P_Calc_New_Mol_Packed Section_Holdup
% Return
% Calculate Packed Section Molar Holdups
% given
% Time, t;
%=====
function [M_Holdup] = P_Calc_New_Mol_Packed_Section_Holdup(t, Ro_l_Density,...
    Mw_MolWeight, M_Holdup_pr, L_flow)
Glob_Decls;

% Set Size
M_Holdup = M_Holdup_pr;

% Calculate Packed Section Molar Holdup [mol]
for i=2:NT+1;
    Fr = ((L_flow (2)*Mw_MolWeight(i))/(Ro_l_Density(i)*A*3600)^2)*(ap/(g*(e_porosity^4.65)));
    ho = 0.555 * ((Fr)^(1/3));
    M_Holdup(i) = (A * ho * Ro_l_Density(i)*z_total)/NT/(Mw_MolWeight(i));
end;

%end P_Calc_New_Mol_Packed Section_Holdup
%=====
% P_Calc_New_Mol_Drum_Holdup
% Return
% Calculate new Reflux-Drum molar holdup
% given
% Time, t;
%=====
function [M_Holdup] = P_Calc_New_Mol_Drum_Holdup(t, M_Holdup_pr)
Glob_Decls;

% Set Size
M_Holdup = M_Holdup_pr;

% Reflux Drum Molar holdup is constant
M_Holdup(NT+2) = M_Holdup_pr(NT+2);

%end P_Calc_New_Mol_Drum_Holdup
%=====
% P_Calc_New_Apprx_Deriv
% Return
% Calculate Approximated derivatives
% given
% Time, t;
%=====
function [Del_M_Holdup, Del_H_l_Enthalpy, Del_M_Hl] = P_Calc_New_Apprx_Deriv(t,...
    M_Holdup, H_l_Enthalpy, t_prv, M_Holdup_prv, H_l_Enthalpy_prv)
Glob_Decls;

% Set Sizes
Del_M_Holdup = zeros(size(M_Holdup));
Del_H_l_Enthalpy = zeros(size(H_l_Enthalpy));
Del_M_Hl = zeros(size(M_Holdup));

% Calculate Step Size
del_t = t - t_prv;
% d(M_Holdup)/dt approximated by forward differentiation
Del_M_Holdup = (M_Holdup - M_Holdup_prv) / del_t;

```

```

for i = 2: NT+1;
    Del_M_Holdup(i) = 0;
end

% d(H_l_Enthalpy)/dt approximated by forward differentiation
Del_H_l_Enthalpy = (H_l_Enthalpy - H_l_Enthalpy_prv) / del_t;
for i = 2: NT+1;
    Del_H_l_Enthalpy(i) = 0;
end

% d(M_Holdup*H_l_Enthalpy)/dt approximated by forward differentiation
Del_M_Hl = (M_Holdup.*H_l_Enthalpy - M_Holdup_prv.*H_l_Enthalpy_prv) / del_t;
for i = 2: NT+1;
    Del_M_Hl(i) = 0;
end

%end P_Calc_New_Apprx_Deriv
=====
% P_Calc_LV_for_Total_Reflux
% Return
%     Calculates Liquid and Vapor Flow Rates for Total Reflux Condition
%     or for D=0
% given
%     Time, t;
=====
function [D_DistillRate, L_flow, V_flow] = P_Calc_LV_for_Total_Reflux(t,...
    H_l_Enthalpy, H_v_Enthalpy, M_Holdup, Del_M_Holdup, Del_H_l_Enthalpy,...
    Del_M_Hl, Q_Boiler, R_Ratio_inv)
Glob_Decls;

% Set Sizes
D_DistillRate = zeros(size(1,1));
L_flow = zeros(size(M_Holdup));
V_flow = zeros(size(M_Holdup));

% Calculate Distillate Flow Rate
D_DistillRate = 0.0;

% Calculate Liquid Flow Rate from Reflux Drum
sum_Del_M_Holdup = sum( Del_M_Holdup(2:NT+1) );
sum_Del_M_Hl = sum( Del_M_Hl(2:NT+1) );
L_flow(NT+2) = ( Q_Boiler - H_l_Enthalpy(1)*Del_M_Holdup(1) - M_Holdup(1)...
    *Del_H_l_Enthalpy(1) - sum_Del_M_Hl ) / ( H_v_Enthalpy(NT+1) - H_l_Enthalpy(NT+2));

% Calculate Vapor Flow Rate from Top Section
V_flow(NT+1) = L_flow(NT+2);

% Calculate Other Vapor and Liquid Flow Rates
%(Since equimolar counter diffusion assumed)
for i= NT+1:-1:2;
    V_flow(i-1) = V_flow(NT+1);
    L_flow(i) = L_flow(NT+2);
end

%end P_Calc_LV_for_Total_Reflux
=====
% P_Calc_LV_for_Finite_Reflux
% Return
%     Calculates Liquid and Vapor Flow Rates for Distillate Withdrawal
%     or for D!=0 or D Different than Zero.
% given
%     Time, t;
=====
function [D_DistillRate, L_flow, V_flow] = P_Calc_LV_for_Finite_Reflux(t,...
    H_l_Enthalpy, H_v_Enthalpy, M_Holdup, Del_M_Holdup, Del_H_l_Enthalpy, ...
    Del_M_Hl, Q_Boiler, R_Ratio_inv)
Glob_Decls;

% Set Sizes
D_DistillRate = zeros(size(1,1));
L_flow = zeros(size(M_Holdup));
V_flow = zeros(size(M_Holdup));

% Calculate Distillate Flow Rate
R = 1 / R_Ratio_inv;
Dummy1 = sum(Del_M_Hl(1:NT+1));
D_DistillRate = ( Q_Boiler - Dummy1 ) / ( (R+1)*H_v_Enthalpy(NT+1) - R*H_l_Enthalpy(NT+2) );

% Calculate Vapor Flow Rate from Top Section
V_flow(NT+1) = D_DistillRate * (R+1);
% Calculate Liquid Flow Rate from Reflux Drum
L_flow(NT+2) = D_DistillRate * R;

% Calculate Other Vapor and Liquid Flow Rates
%(Since equimolar counter diffusion assumed)
for i=NT+1:-1:2;
    V_flow(i-1) = V_flow(i);
    L_flow(i) = L_flow(i+1);
end;

%end P_Calc_LV_for_Finite_Reflux
=====
% P_Calc_Tanks
% Return
%     Storage Tank Holdups and Compositions
=====
function [X, M] = P_Calc_Tanks(t, DeltaT, X, M, X_Drum, D_Rate, Active);

```



```

if Active~=0;
    % Calculate increase in holdup
    deltaM = D_Rate*DeltaT;

    % Calculate tank's composition
    if ((deltaM == 0) & M(Active, 1)==0)
        X(Active, :) = zeros(size(X(Active, :)));
    else
        X(Active, :) = (X(Active, :)* M(Active, 1)+X_Drum(1, :)*deltaM)/ (M(Active,1)+deltaM);
    end;

    % Calculate tank's current holdup
    M(Active, 1) = M(Active, 1) + D_Rate*DeltaT;
end;

%end P_Calc_Tanks
=====
% P_Calc_New_Distilled_Amount
% Return
% Calculate New Amount of Product Distilled
% given
% Time, t;
=====
function [M_Distilled] = P_Calc_New_Distilled_Amount(t, t_prv, D_DistillRate, M_Distilled_pr)
Glob_Decls;

% Set Size
M_Distilled = 0.0;

% Calculate Amount of Product Distilled
M_Distilled = M_Distilled_pr + D_DistillRate*(t-t_prv);

%end P_Calc_New_Distilled_Amount
=====
% P_Calc_New_Mol_Still_Holdup
% Return
% Calculate Still Pot Holdup
% given
% Time, t;
=====
function [M_Holdup] = P_Calc_New_Mol_Still_Holdup(t, M_Holdup_pr, M_Distilled);
Glob_Decls;

% Set Sizes
M_Holdup = M_Holdup_pr;

% Instantaneous Still Pot Total Holdup Amount [moles/hour]
M_Holdup(1) = M_Feed - sum(M_Holdup_pr(2:NT+1)) - M_Holdup_pr(NT+2) - M_Distilled;

%end P_Calc_New_Mol_Still_Holdup
% -----End Real Plant Simulation Functions----- %
% ##### %
% ##### %
% ----- Controller Functions ----- %
% ----- %
=====
% INIT_CONTROL (don't MODIFY GLOBAL VARIABLES not owned by this function)
% perform
% Initialize controller
% given
% All Global variables
% output
% any output required
=====
function [SetPoints, Num_Oper_Stage, DistillProfile] = INIT_CONTROL
Glob_Decls;

% Controller Set Points of Product Specifications
SetPoints = zeros(NC,1);
SetPoints = [ 0.9; 0.8; 0.69];

% Number of different operation stage
Num_Oper_Stage = 5;

% Distillate Flow rate values of different operation stages (Products and/or Slopcuts)
DistillProfile = zeros(Num_Oper_Stage,1);
%DistillProfile = [];

%end INIT_CONTROL
=====
% CONTROL (don't MODIFY GLOBAL VARIABLES not owned by this function)
% perform
% Controls the system
% given
% Current Time, t; All Global variables at time t;
% output
% any output required
=====
function [CONT_QBoiler, CONT_RRatio_inv, Tank_Activated] = CONTROL(t, X_frac, ...
    Q_Boiler_pr, CONT_SetPoints, CONT_Num_Oper_Stage, CONT_DistillProfile)
Glob_Decls;
R_Ratio = zeros(1);
% Find new Reflux ratio (L0/D)
if (t>= 0 & t < 1.0)
    CONT_RRatio_inv = 0.0;
    Tank_Activated = 1;
elseif (t >= 1.0 & t < 3.1134)

```

```

R_Ratio = 0.889/ (1.0 - 0.889);
CONT_RRatio_inv = 1.0 / R_Ratio;
Tank_Activated = 1;
elseif (t>=3.1134 & t< 5.8782)
R_Ratio = 0.825/ (1.0 - 0.825);
CONT_RRatio_inv = 1.0 / R_Ratio;
Tank_Activated = 2;
elseif (t>=5.8782 & t<6.6039)
R_Ratio = 0.8/ (1.0 - 0.8);
CONT_RRatio_inv = 1.0 / R_Ratio;
Tank_Activated = 3;
elseif (t >= 6.6039 )
R_Ratio = 0.8/ (1.0 - 0.8);
CONT_RRatio_inv = 1.0 / R_Ratio;
Tank_Activated = 3;
end;

% Find Reboiler load (J/hour)
% Constant Reboiler Load
CONT_QBoiler = Q_Boiler_pr;

%end CONTROL
=====
% CONTROL_real (don't MODIFY GLOBAL VARIABLES not owned by this function)
% perform
% Controls the system
% given
% Current Time, t; All Global variables at time t;
% output
% any output required
=====
function [Q_Boiler, R_Ratio_inv, Tank_Active, CONT_Curr_Stage_new] = CONTROL_real(t,...
X_frac, Q_Boiler_pr, R_Ratio_inv_pr, CONT_SetPoints, CONT_Num_Oper_Stage,...
CONT_DistillProfile, Tank_Active_prv, X_tank, CONT_Curr_Stage)
Glob_Decls;
R_Ratio = zeros(1);
% Find new Reflux ratio (L0/D)

if (CONT_Curr_Stage==0) %----- Total Reflux operation
if (X_frac(Nt+2,1)<CONT_SetPoints(1,1))
R_Ratio_inv = 0.0;
Tank_Active = 0;
CONT_Curr_Stage = 0;
elseif (X_frac(Nt+2,1)>=CONT_SetPoints(1,1))
'0->1'
%pause;
CONT_Curr_Stage = 1;
end;
end;

if (CONT_Curr_Stage==1) %----- 1st product-cut distillation to 1st product-cut tank
if ((X_tank(1,1)==0) | (X_tank(1,1)>=CONT_SetPoints(1,1)))
if ((R_ratio_opt == 1) | (R_ratio_opt == 0))
R_Ratio = -1;
Tank_Active = -1;
else
% R_Ratio = R_ratio_opt / (1.0 - R_ratio_opt);
R_Ratio = 0.889/ (1.0 - 0.889);
Tank_Active = 1;
end;
CONT_Curr_Stage = 1;
elseif (X_tank(1,1)<CONT_SetPoints(1,1))
'1->2'
% pause;
CONT_Curr_Stage = 2;
end;
end;

if (CONT_Curr_Stage==2) %----- 1st slop-cut distillation to 1st slop-cut tank
if (X_frac(Nt+2,2)<CONT_SetPoints(2,1))
if ((R_ratio_opt == 1) | (R_ratio_opt == 0))
R_Ratio = -1;
Tank_Active = -1;
else
% R_Ratio = R_ratio_opt / (1.0 - R_ratio_opt);
R_Ratio = 0.825/ (1.0 - 0.825);
Tank_Active = 2;
end;
CONT_Curr_Stage = 2;
elseif (X_frac(Nt+2,2)>=CONT_SetPoints(2,1))
'2->3'
%pause;
CONT_Curr_Stage = 3;
end;
end;

if (CONT_Curr_Stage==3) %----- 2nd product-cut distillation to 2nd product-cut tank
if ((X_tank(3,2)==0) | (X_tank(3,2)>=CONT_SetPoints(2,1)))
if ((R_ratio_opt == 1) | (R_ratio_opt == 0))
R_Ratio = -1;
Tank_Active = -1;
else
R_Ratio = R_ratio_opt / (1.0 - R_ratio_opt);
% R_Ratio = 0.8 / (1.0 - 0.8);
Tank_Active = 3;
end;
end;

```

```

        CONT_Curr_Stage = 3;
    elseif (X_tank(3,2)<CONT_SetPoints(2,1))
        '3->4'
        %pause;
        CONT_Curr_Stage = 4;
    end;
end;

if (CONT_Curr_Stage==4) %----- 2nd slop-cut distillation to 2nd slop-cut tank
    if (X_frac(1,3)<CONT_SetPoints(3,1))
        if ((R_ratio_opt == 1) | (R_ratio_opt == 0))
            R_Ratio = -1;
            Tank_Active = -1;
        else
            R_Ratio = R_ratio_opt / (1.0 - R_ratio_opt);
            Tank_Active = -1;
        end;
        CONT_Curr_Stage = 4;
    elseif (X_frac(1,3)>=CONT_SetPoints(3,1))
        '4->5'
        %pause;
        CONT_Curr_Stage = 5;
    end;
end;
if (X_frac(1,3)>=CONT_SetPoints(3,1))
    '4->5'
    %pause;
    CONT_Curr_Stage = 5;
end;
if (CONT_Curr_Stage==5) %----- Distillation stops
    R_Ratio = 1/R_Ratio_inv_pr;
    Tank_Active = -1;
end;

% Keep Current Stage #
CONT_Curr_Stage_new = CONT_Curr_Stage;

% Convert Reflux ratio (L0/D) to One Over Reflux ratio (D/L0)
if (R_Ratio==-1)
    R_Ratio_inv = 0;
else
    R_Ratio_inv = 1.0 / R_Ratio;
end;

% Find Reboiler load (J/hour)
% Constant Reboiler Load
Q_Boiler = Q_Boiler_pr;

%end CONTROL_real

% ----- %
% -----End Controller Functions----- %
% ##### %

% ##### %
% -----Simulation loop control user interface functions-----%
% -----%
%=====
% write_plant_to_scr
%=====
function write_plant_to_scr(t, X_frac, R_Ratio_inv)
Glob_Decls;

% Reflux Drum Liquid Composition
data = X_frac(NT+2,:);
y = X_frac(1,3);
l = prod(size(data));
formats = '';
for i=1:l;
    formats = [formats ' %f'];
end;
fprintf(['%9.4f' formats '\n'], t, reshape(data, 1, prod(size(data))), R_Ratio_inv,y);

%end write_plant_to_scr

%=====
% write_plant_to_file
%=====
function write_plant_to_file(t, X_frac, Y_frac, Temp, M_Holdup, L_flow, V_flow)
Glob_Decls;

% Liquid Profile File
l = prod(size(X_frac));
formats = '';
for i=1:l;
    formats = [formats ' %f'];
end;
fprintf(FID_lprofile, ['%9.4f' formats '\n'], t, reshape(X_frac, 1, l));

% Vapor Profile File
l = prod(size(Y_frac));
formats = '';
for i=1:l;
    formats = [formats ' %f'];
end;
fprintf(FID_vprofile, ['%9.4f' formats '\n'], t, reshape(Y_frac, 1, l));

```

```

% Temperature Profile File
l = prod(size(Temp));
formats = '';
for i=1:l; formats = [formats ' ' %f'];
end;
fprintf(FID_tprofile, ['%9.4f' formats '\n'], t, reshape(Temp, 1, l));

% Holdup Profile File
l = prod(size(M_Holdup));
formats = '';
for i=1:l;
    formats = [formats ' ' %f'];
end;
fprintf(FID_holdup, ['%9.4f' formats '\n'], t, reshape(M_Holdup, 1, l));

% Liquid and Vapor Flowrate Profile File
l = prod(size(L_flow)) + prod(size(V_flow));
formats = '';
for i=1:l;
    formats = [formats ' ' %f'];
end;
fprintf(FID_lvflow, ['%9.4f' formats '\n'], t, reshape([L_flow; V_flow], 1, l));

%end write_plant_to_file
%=====
% write_estcont_to_file : Write estimator and controller data to file
%=====
function write_estcont_to_file(t, CONT_QBoiler, CONT_RRatio_inv);
Glob_Decls;

% Controller Outputs file
l = prod(size(CONT_QBoiler)) + prod(size(CONT_RRatio_inv));
formats = ''; for i=1:l; formats = [formats ' ' %f']; end;
fprintf(FID_control, ['%9.4f ' formats '\n'], t, CONT_QBoiler, CONT_RRatio_inv);

%end write_estcont_to_file

%=====
% write_tank_to_file : Write tank data to file
%=====
function write_tank_to_file(t, X_actual, M_actual);
Glob_Decls;

l = prod(size(X_actual)) + prod(size(M_actual));
formats = ''; for i=1:l; formats = [formats ' ' %f']; end;
fprintf(FID_tank, ['%9.4f ' formats '\n'], t, reshape(X_actual, 1, prod(size(X_actual))),...
    reshape(M_actual, 1, prod(size(M_actual))));

%end write_tank_to_file

%=====
% write_opt_to_file : Write optimization data to file
%=====
function write_opt_to_file(t, R_ratio_opt, Tank1_Holdup, Tank2_Holdup, Tank3_Holdup, ...
    Tank4_Holdup, M_Holdup, cap_fact);
Glob_Decls;

l = prod(size(R_ratio_opt)) + prod(size(Tank1_Holdup)) + prod(size(Tank2_Holdup)) + ...
    prod(size(Tank3_Holdup)) + prod(size(Tank4_Holdup)) + prod(size(M_Holdup)) + prod(size(cap_fact));
formats = ''; for i=1:l; formats = [formats ' ' %f']; end;
fprintf(FID_opt, ['%9.4f ' formats '\n'], t, reshape(R_ratio_opt, 1, prod(size(R_ratio_opt))),...
    reshape(Tank1_Holdup, 1, prod(size(Tank1_Holdup))),...
    reshape(Tank2_Holdup, 1, prod(size(Tank2_Holdup))),...
    reshape(Tank3_Holdup, 1, prod(size(Tank3_Holdup))),...
    reshape(Tank4_Holdup, 1, prod(size(Tank4_Holdup))),...
    reshape(M_Holdup, 1, prod(size(M_Holdup))),...
    reshape(cap_fact, 1, prod(size(cap_fact))));
%end write_opt_to_file
% -----End Simulation loop control user interface functions----- %
% ##### %

```

## C.2 Optimization Code

### C.2.1 Optimizer.m

```

%=====
%      Date      By      Explanation
%=====
%      2006      Bahar      Original Code
%=====

function Optimum_R = OptimizeR(Rinitial)

x0 = Rinitial; % Make a starting guess at the solution
%options = optimset('LargeScale','off');
options = optimset('LargeScale','off','TolX',1e-3);
[x, fval] = fmincon(@Batch_obj,x0,[],[],[],[],[],[],@Batch_con,options)
%[x, fval] = fmincon(@Batch_obj,x0,[],[],[],[],0,1,[],options)

Optimum_R = x;
save
#####

```

## C.2.2 Batch\_con.m

```
=====
%      Date      By      Explanation
% =====
%      2006      Bahar      Original Code

function [c, ceq] = Batch_con(x)
% Nonlinear inequality constraints
c = [ -x(1) ; x(1)-1 ];
% Nonlinear equality constraints
ceq = [];
#####
```

## C.2.3 Batch\_obj.m

```
=====
%      Date      By      Explanation
% =====
%      2006      Bahar      Original Code
%      2006      Ceylan      Modified to packed distillation

function [CAPfromOBJ] = Batch_obj(RfromOPT)

% Include all global variables
Glob_Decls;

% Run simulation
CAPfromOBJ = plant_file_packed(RfromOPT);

%CAPfromOBJ = CAPtoOPT;
CAPfromOBJ = -cap_fact;
#####
```

## C.3 Thermodynamic Library

### C.3.1 thermo\_Init.m

```
=====
%      Date      By      Explanation
% =====
%      2002      Yildiz      Original Code

function thermo_Init(check_input_parameters)
% function thermo_Init(check_input_parameters)
% Thermophysical and physical property calculation MEX File Interface
%
% ----- Initialization routine -----
% if check_input_parameters = 1 then initialization routine writes the
% input parameters read from 'plant_data.dat' to 'plant_data_check.dat'

thermo_LIBRARY('init',check_input_parameters);
```

### C.3.2 thermo\_Equilibrium\_Hydrocarbons.m

```
=====
%      Date      By      Explanation
% =====
%      2002      Yildiz      Original Code
%      2006      Ceylan      Modification to Packed Dist.

function [Tequi, y] = thermo_Equilibrium_Hydrocarbons(T,P,x)
% function [Tequi, y] = thermo_Equilibrium(T,P,x)
% Thermophysical and physical property calculation MEX File Interface
%
% ----- Equilibrium routine -----
% [Tequi, y] : Equilibrium temperature(K), Equilibrium vapor phase fractions(mol/mol)
% (T,P,x) : Initial Equilibrium temperature guess(K), Pressure(Pa), Liquid phase fractions(mol/mol)

[Tequi, y] = thermo_LIBRARY('equilibrium_hydrocarbons',T,P,x);
```

### C.3.3 thermo\_Equilibrium\_Polar.m

```
=====
%      Date      By      Explanation
=====
%      2002      Yıldız      Original Code
%      2006      Ceylan      Modification to Packed Dist.

function [Tequi, y] = thermo_Equilibrium_Polar(T,P,x)
% function [Tequi, y] = thermo_Equilibrium(T,P,x)
% Thermophysical and physical property calculation MEX File Interface
%
% ----- Equilibrium routine -----
% [Tequi, y] : Equilibrium temperature(K), Equilibrium vapor phase fractions(mol/mol)
% (T,P,x) : Initial Equilibrium temperature guess(K), Pressure(Pa), Liquid phase fractions(mol/mol)

[Tequi, y] = thermo_LIBRARY('equilibrium_polar',T,P,x);
```

### C.3.4 thermo\_Enthalpy.m

```
=====
%      Date      By      Explanation
=====
%      2002      Yıldız      Original Code
%      2006      Ceylan      Modification to Packed Dist.

function [hl,hv] = thermo_Enthalpy(T,P,x,y)
% function [hl,hv] = thermo_Enthalpy(T,P,x,y)
% Thermophysical and physical property calculation MEX File Interface
%
% ----- Enthalpy routine -----
% [hl,hv] : Liquid and Vapor phase specific enthalpy (J/mol)
% (T,P,x,y) : Initial Equilibrium temperature guess(K), Pressure(Pa),
%           Liquid phase fractions(mol/mol), vapor phase fractions(mol/mol)

[hl,hv] = thermo_LIBRARY('enthalpy',T,P,x,y);
```

### C.3.5 thermo\_L\_Density.m

```
=====
%      Date      By      Explanation
=====
%      2002      Yıldız      Original Code
%      2006      Ceylan      Modification to Packed Dist.

function [mwa, densa] = thermo_L_Density(T,P,x)
% function [mwa, densa] = thermo_L_Density(T,P,x)
% Thermophysical and physical property calculation MEX File Interface
%
% ----- Density routine -----
% [mwa, densa] : Average molecular weight (kg/mol), Average liquid phase density (kg/m3)
% (T,P,x) : Initial Equilibrium temperature guess(K), Pressure(Pa), Liquid phase fractions(mol/mol)

[mwa, densa] = thermo_LIBRARY('l_density',T,P,x);
```

### C.3.6 thermo\_G\_Density.m

```
=====
%      Date      By      Explanation
=====
%      2002      Yıldız      Original Code
%      2006      Ceylan      Modification to Packed Dist.

function [mwa, densa] = thermo_G_Density(T,P,x)
% function [mwa, densa] = thermo_G_Density(T,P,x)
% Thermophysical and physical property calculation MEX File Interface
%
% ----- Density routine -----
% [mwa, densa] : Average molecular weight (kg/mol), Average vapor phase density (kg/m3)
% (T,P,x) : Initial Equilibrium temperature guess(K), Pressure(Pa), Liquid phase fractions(mol/mol)

[mwa, densa] = thermo_LIBRARY('g_density',T,P,x);
```

### C.3.7 thermo\_LIBRARY.f

```

C -----
C -----
C      MEX File Gateway implementation for Plant_Subroutines
C      Date : 08-05-2001
C      by Uğur YILDIZ
C -----
C -----
      subroutine mexFunction(nlhs, plhs, nrhs, prhs)
         include 'thermo_LIBRARY.h'
         include 'parameter.h'
         integer plhs(*), prhs(*)           ! pointer to left-hand and right-hand side variables
         integer nlhs, nrhs                ! # of variables in plhs, prhs
         integer mxCreateFull, mxGetString ! mx Functions declarations
         integer mxGetM, mxGetN, mxIsNumeric, mxIsString ! mx Functions declarations
         integer m, n, size, status, alloc_err ! Dummy variables
         integer Func_name_ptr             ! Function name fortran pointers
         character*100 Func_name           ! Function name for fortran use
C ----- Input fortran pointers
         integer Input1_pr, Input2_pr, Input3_pr, Input4_pr, Input5_pr, Input6_pr, x_pr, y_pr, z_pr
C ----- Output fortran pointers
         integer Output1_pr, Output2_pr, Output3_pr, Output4_pr, Output5_pr, Output6_pr
C ----- Input arguments for fortran use
         integer,allocatable, dimension (:): int_Input1, int_Input2, int_Input3, int_Input4, int_Input5, int_Input6
         real*8,allocatable, dimension (:): real_Input1, real_Input2, real_Input3, real_Input4, real_Input5, real_Input6
         integer Input1_sz,Input2_sz,Input3_sz,Input4_sz,Input5_sz,Input6_sz
C ----- Output arguments for fortran use
         integer,allocatable, dimension (:): int_Output1, int_Output2, int_Output3, int_Output4, int_Output5, int_Output6
         real*8,allocatable, dimension (:): real_Output1, real_Output2, real_Output3, real_Output4, real_Output5, real_Output6
         integer Output1_sz,Output2_sz,Output3_sz,Output4_sz,Output5_sz,Output6_sz
         real*8 x, y(3,3), z(3,3), ugur
C ----- Check for at least one function is requested.
         if (nrhs .lt. 1) then
            call mexErrMsgTxt('Not a proper function selected. - [thermo_LIBRARY.dll]')
         endif
         if (mxIsString(prhs(1)) .ne. 1) then
            call mexErrMsgTxt('Function name parameter must be a valid string. - [thermo_LIBRARY.dll]')
         endif
         m = mxGetM(prhs(1))
         n = mxGetN(prhs(1))
         if (m .ne. 1) then
            call mexErrMsgTxt('Function name parameter must be a row vector. - [thermo_LIBRARY.dll]')
         endif
C ----- Call the requested function.
C      Get the string contents (dereference the input integer).
         status = mxGetString(prhs(1),Func_name,100)
C      Check if mxGetString is successful.
         if (status .ne. 0) then
            call mexErrMsgTxt('String length must be less than 100. - [thermo_LIBRARY.dll]')
         endif
C ----- ! Call initialization function
         if (Func_name.eq.'init') then
            status = 1
            if (nrhs .ne. 2) then
               call mexErrMsgTxt('One input (number of components) is required for the initialization. - (init) [thermo_LIBRARY.dll]')
            endif
            if (mxIsNumeric(prhs(2)) .ne. 1) call mexErrMsgTxt('Input #1 is not a numeric. - (init) [thermo_LIBRARY.dll]')
            m = mxGetM(prhs(2))
            n = mxGetN(prhs(2))
            if (n .ne. 1 .or. m .ne. 1) call mexErrMsgTxt('Input #1 is not a scalar. - (init) [thermo_LIBRARY.dll]')

```

```

Input1_pr = mxGetPr(prhs(2))
status = 0
allocate (int_Input1(1),STAT = alloc_err)
status = status + alloc_err
if (status .ne. 0) then
  if (allocated(int_Input1)) deallocate(int_Input1)
  call mexErrMsgTxt('Memory allocation error. - (init) [thermo_LIBRARY.dll]')
endif
call mxCopyPtrToInteger4(Input1_pr,int_Input1,1)
status = 1
call init(int_Input1,status)
if (status.eq.0) then
  call mexPrintf('thermo_LIBRARY is initialized. - (init) [thermo_LIBRARY.dll]')
else
  call mexErrMsgTxt('thermo_LIBRARY can not be initialized. - (init) [thermo_LIBRARY.dll]')
endif
c ----- ! Call enthalpy function
elseif ((Func_name.eq.'enthalpy') .and. (lib_Inited.eq.1)) then
  if (nrhs .ne. 5) then
    call mexErrMsgTxt('Four inputs (T,P,x,y) is required. - (enthalpy) [thermo_LIBRARY.dll]')
  elseif (nlhs .ne. 2) then
    call mexErrMsgTxt('Two outputs (liquid and vapor enthalpies) are required. - (enthalpy) [thermo_LIBRARY.dll]')
  endif
  if (mxIsNumeric(prhs(2)) .ne. 1) call mexErrMsgTxt('Input #1 is not a numeric. - (enthalpy) [thermo_LIBRARY.dll]')
  if (mxIsNumeric(prhs(3)) .ne. 1) call mexErrMsgTxt('Input #2 is not a numeric. - (enthalpy) [thermo_LIBRARY.dll]')
  if (mxIsNumeric(prhs(4)) .ne. 1) call mexErrMsgTxt('Input #3 is not a numeric. - (enthalpy) [thermo_LIBRARY.dll]')
  if (mxIsNumeric(prhs(5)) .ne. 1) call mexErrMsgTxt('Input #4 is not a numeric. - (enthalpy) [thermo_LIBRARY.dll]')
  m = mxGetM(prhs(2))
  n = mxGetN(prhs(2))
  Input1_sz = m*n
  if (n .ne. 1 .or. m .ne. 1) call mexErrMsgTxt('Input #1 is not a scalar. - (enthalpy) [thermo_LIBRARY.dll]')
  m = mxGetM(prhs(3))
  n = mxGetN(prhs(3))
  Input2_sz = m*n
  if (n .ne. 1 .or. m .ne. 1) call mexErrMsgTxt('Input #2 is not a scalar. - (enthalpy) [thermo_LIBRARY.dll]')
  m = mxGetM(prhs(4))
  n = mxGetN(prhs(4))
  Input3_sz = m*n
  if (n .ne. nj .or. m .ne. 1) call mexErrMsgTxt('Input #3 is not a NC-element row vector. - (enthalpy) [thermo_LIBRARY.dll]')
  m = mxGetM(prhs(5))
  n = mxGetN(prhs(5))
  Input4_sz = m*n
  if (n .ne. nj .or. m .ne. 1) call mexErrMsgTxt('Input #4 is not a NC-element row vector. - (enthalpy) [thermo_LIBRARY.dll]')
  Input1_pr = mxGetPr(prhs(2))
  Input2_pr = mxGetPr(prhs(3))
  Input3_pr = mxGetPr(prhs(4))
  Input4_pr = mxGetPr(prhs(5))
  plhs(1) = mxCreateFull(1,1,0)
  plhs(2) = mxCreateFull(1,1,0)
  Output1_pr = mxGetPr(plhs(1))
  Output2_pr = mxGetPr(plhs(2))
  status = 0
  allocate (real_Input1(1),STAT = alloc_err)
  status = status + alloc_err
  allocate (real_Input2(1),STAT = alloc_err)
  status = status + alloc_err
  allocate (real_Input3(nj),STAT = alloc_err)
  status = status + alloc_err
  allocate (real_Input4(nj),STAT = alloc_err)
  status = status + alloc_err
  allocate (real_Output1(1),STAT = alloc_err)
  status = status + alloc_err

```



```

allocate (real_Output2(1),STAT = alloc_err)
status = status + alloc_err
if (status .ne. 0) then
  if (allocated(real_Input1)) deallocate(real_Input1)
  if (allocated(real_Input2)) deallocate(real_Input2)
  if (allocated(real_Input3)) deallocate(real_Input3)
  if (allocated(real_Input4)) deallocate(real_Input4)
  if (allocated(real_Output1)) deallocate(real_Output1)
  if (allocated(real_Output2)) deallocate(real_Output2)
  call mexErrMsgTxt('Memory allocation error. - (enthalpy) [thermo_LIBRARY.dll]')
endif
call mxCopyPtrToReal8(Input1_pr,real_Input1,Input1_sz)
call mxCopyPtrToReal8(Input2_pr,real_Input2,Input2_sz)
call mxCopyPtrToReal8(Input3_pr,real_Input3,Input3_sz)
call mxCopyPtrToReal8(Input4_pr,real_Input4,Input4_sz)
call enth(real_Input1, real_Input2, real_Input3, real_Input4, real_Output1, real_Output2)
call mxCopyReal8ToPtr(real_Output1,Output1_pr,1)
call mxCopyReal8ToPtr(real_Output2,Output2_pr,1)
c ----- ! Call liquid density function
elseif ((Func_name.eq.'l_density') .and. (lib_Initialized.eq.1)) then
  if (nrhs .ne. 4) then
    call mexErrMsgTxt('Three inputs (T,P,x) is required. - (density) [thermo_LIBRARY.dll]')
  elseif (nrhs .ne. 2) then
    call mexErrMsgTxt('Two outputs (Avg. mol. weight and density) are required. - (density) [thermo_LIBRARY.dll]')
  endif
  if (mxIsNumeric(prhs(2)) .ne. 1) call mexErrMsgTxt('Input #1 is not a numeric. - (density) [thermo_LIBRARY.dll]')
  if (mxIsNumeric(prhs(3)) .ne. 1) call mexErrMsgTxt('Input #2 is not a numeric. - (density) [thermo_LIBRARY.dll]')
  if (mxIsNumeric(prhs(4)) .ne. 1) call mexErrMsgTxt('Input #3 is not a numeric. - (density) [thermo_LIBRARY.dll]')
  m = mxGetM(prhs(2))
  n = mxGetN(prhs(2))
  Input1_sz = m*n
  if (n .ne. 1 .or. m .ne. 1) call mexErrMsgTxt('Input #1 is not a scalar. - (density) [thermo_LIBRARY.dll]')
  m = mxGetM(prhs(3))
  n = mxGetN(prhs(3))
  Input2_sz = m*n
  if (n .ne. 1 .or. m .ne. 1) call mexErrMsgTxt('Input #2 is not a scalar. - (density) [thermo_LIBRARY.dll]')
  m = mxGetM(prhs(4))
  n = mxGetN(prhs(4))
  Input3_sz = m*n
  if (n .ne. nj .or. m .ne. 1) call mexErrMsgTxt('Input #3 is not a NC-element row vector. - (density) [thermo_LIBRARY.dll]')
  Input1_pr = mxGetPr(prhs(2))
  Input2_pr = mxGetPr(prhs(3))
  Input3_pr = mxGetPr(prhs(4))
  plhs(1) = mxCreateFull(1,1,0)
  plhs(2) = mxCreateFull(1,1,0)
Output1_pr = mxGetPr(plhs(1))
Output2_pr = mxGetPr(plhs(2))
status = 0
allocate (real_Input1(1),STAT = alloc_err)
status = status + alloc_err
allocate (real_Input2(1),STAT = alloc_err)
status = status + alloc_err
allocate (real_Input3(nj),STAT = alloc_err)
status = status + alloc_err
allocate (real_Output1(1),STAT = alloc_err)
status = status + alloc_err
allocate (real_Output2(1),STAT = alloc_err)
status = status + alloc_err
if (status .ne. 0) then
  if (allocated(real_Input1)) deallocate(real_Input1)
  if (allocated(real_Input2)) deallocate(real_Input2)
  if (allocated(real_Input3)) deallocate(real_Input3)

```

```

        if (allocated(real_Output1)) deallocate(real_Output1)
        if (allocated(real_Output2)) deallocate(real_Output2)
        call mexErrMsgTxt('Memory allocation error. - (density) [thermo_LIBRARY.dll]')
    endif
    call mxCopyPtrToReal8(Input1_pr,real_Input1,Input1_sz)
    call mxCopyPtrToReal8(Input2_pr,real_Input2,Input2_sz)
    call mxCopyPtrToReal8(Input3_pr,real_Input3,Input3_sz)
c      subroutine pr_dens_l(t,p,x,mwa,densa)
        call pr_dens_l(real_Input1, real_Input2, real_Input3, real_Output1, real_Output2)
        call mxCopyReal8ToPtr(real_Output1,Output1_pr,1)
        call mxCopyReal8ToPtr(real_Output2,Output2_pr,1)
c ----- ! Call vapor density function
        elseif ((Func_name.eq.'g_density') .and. (lib_Inited.eq.1)) then
            if (nrhs .ne. 4) then
                call mexErrMsgTxt('Three inputs (T,P,x) is required. - (density) [thermo_LIBRARY.dll]')
            elseif (nlhs .ne. 2) then
                call mexErrMsgTxt('Two outputs (Avg. mol. weight and density) are required. - (density) [thermo_LIBRARY.dll]')
            endif
            if (mxIsNumeric(prhs(2)) .ne. 1) call mexErrMsgTxt('Input #1 is not a numeric. - (density) [thermo_LIBRARY.dll]')
            if (mxIsNumeric(prhs(3)) .ne. 1) call mexErrMsgTxt('Input #2 is not a numeric. - (density) [thermo_LIBRARY.dll]')
            if (mxIsNumeric(prhs(4)) .ne. 1) call mexErrMsgTxt('Input #3 is not a numeric. - (density) [thermo_LIBRARY.dll]')
            m = mxGetM(prhs(2))
            n = mxGetN(prhs(2))
            Input1_sz = m*n
            if (n .ne. 1 .or. m .ne. 1) call mexErrMsgTxt('Input #1 is not a scalar. - (density) [thermo_LIBRARY.dll]')
            m = mxGetM(prhs(3))
            n = mxGetN(prhs(3))
            Input2_sz = m*n
            if (n .ne. 1 .or. m .ne. 1) call mexErrMsgTxt('Input #2 is not a scalar. - (density) [thermo_LIBRARY.dll]')
            m = mxGetM(prhs(4))
            n = mxGetN(prhs(4))
            Input3_sz = m*n
            if (n .ne. nj .or. m .ne. 1) call mexErrMsgTxt('Input #3 is not a NC-element row vector. - (density) [thermo_LIBRARY.dll]')
            Input1_pr = mxGetPr(prhs(2))
            Input2_pr = mxGetPr(prhs(3))
            Input3_pr = mxGetPr(prhs(4))
            plhs(1) = mxCreateFull(1,1,0)
            plhs(2) = mxCreateFull(1,1,0)
            Output1_pr = mxGetPr(plhs(1))
            Output2_pr = mxGetPr(plhs(2))
            status = 0
            allocate (real_Input1(1),STAT = alloc_err)
            status = status + alloc_err
            allocate (real_Input2(1),STAT = alloc_err)
            status = status + alloc_err
            allocate (real_Input3(nj),STAT = alloc_err)
            status = status + alloc_err
            allocate (real_Output1(1),STAT = alloc_err)
            status = status + alloc_err
            allocate (real_Output2(1),STAT = alloc_err)
            status = status + alloc_err
            if (status .ne. 0) then
                if (allocated(real_Input1)) deallocate(real_Input1)
                if (allocated(real_Input2)) deallocate(real_Input2)
                if (allocated(real_Input3)) deallocate(real_Input3)
                if (allocated(real_Output1)) deallocate(real_Output1)
                if (allocated(real_Output2)) deallocate(real_Output2)
                call mexErrMsgTxt('Memory allocation error. - (density) [thermo_LIBRARY.dll]')
            endif
            call mxCopyPtrToReal8(Input1_pr,real_Input1,Input1_sz)
            call mxCopyPtrToReal8(Input2_pr,real_Input2,Input2_sz)
            call mxCopyPtrToReal8(Input3_pr,real_Input3,Input3_sz)

```

```

c      subroutine pr_dens_g(t,p,x,mwa,densa)
      call pr_dens_g(real_Input1, real_Input2, real_Input3, real_Output1, real_Output2)
      call mxCopyReal8ToPtr(real_Output1,Output1_pr,1)
      call mxCopyReal8ToPtr(real_Output2,Output2_pr,1)
c----- ! Call equilibrium function_hydrocarbons
      elseif ((Func_name.eq.'equilibrium_hydrocarbons') .and. (lib_Inited.eq.1)) then
      if (nrhs .ne. 4) then
      call mexErrMsgTxt('Three inputs (T, P, x) is required. - (equilibrium) [thermo_LIBRARY.dll]')
elseif (nrhs .ne. 2) then
      call mexErrMsgTxt('Two outputs (T and vapor comp.) are required. - (equilibrium) [thermo_LIBRARY.dll]')
endif
      if (mxIsNumeric(prhs(2)) .ne. 1) call mexErrMsgTxt('Input #1 is not a numeric. - (equilibrium) [thermo_LIBRARY.dll]')
      if (mxIsNumeric(prhs(3)) .ne. 1) call mexErrMsgTxt('Input #2 is not a numeric. - (equilibrium) [thermo_LIBRARY.dll]')
      if (mxIsNumeric(prhs(4)) .ne. 1) call mexErrMsgTxt('Input #3 is not a numeric. - (equilibrium) [thermo_LIBRARY.dll]')
      m = mxGetM(prhs(2))
      n = mxGetN(prhs(2))
      Input1_sz = m*n
      if (n .ne. 1 .or. m .ne. 1) call mexErrMsgTxt('Input #1 is not a scalar. - (equilibrium) [thermo_LIBRARY.dll]')
      m = mxGetM(prhs(3))
      n = mxGetN(prhs(3))
      Input2_sz = m*n
      if (n .ne. 1 .or. m .ne. 1) call mexErrMsgTxt('Input #2 is not a scalar. - (equilibrium) [thermo_LIBRARY.dll]')
      m = mxGetM(prhs(4))
      n = mxGetN(prhs(4))
      Input3_sz = m*n
      if (n .ne. nj .or. m .ne. 1)
& call mexErrMsgTxt('Input #3 is not a NC-element row vector. - (equilibrium) [thermo_LIBRARY.dll]')
      Input1_pr = mxGetPr(prhs(2))
      Input2_pr = mxGetPr(prhs(3))
      Input3_pr = mxGetPr(prhs(4))
      plhs(1) = mxCreateFull(1,1,0)
      plhs(2) = mxCreateFull(1,nj,0)
      Output1_pr = mxGetPr(plhs(1))
      Output2_pr = mxGetPr(plhs(2))
      status = 0
      allocate (real_Input1(1),STAT = alloc_err)
      status = status + alloc_err
      allocate (real_Input2(1),STAT = alloc_err)
      status = status + alloc_err
      allocate (real_Input3(nj),STAT = alloc_err)
      status = status + alloc_err
      allocate (real_Output1(1),STAT = alloc_err)
      status = status + alloc_err
      allocate (real_Output2(nj),STAT = alloc_err)
      status = status + alloc_err
      if (status .ne. 0) then
      if (allocated(real_Input1)) deallocate(real_Input1)
      if (allocated(real_Input2)) deallocate(real_Input2)
      if (allocated(real_Input3)) deallocate(real_Input3)
      if (allocated(real_Output1)) deallocate(real_Output1)
      if (allocated(real_Output2)) deallocate(real_Output2)
      call mexErrMsgTxt('Memory allocation error. - (equilibrium) [thermo_LIBRARY.dll]')
      endif
      call mxCopyPtrToReal8(Input1_pr,real_Input1,Input1_sz)
      call mxCopyPtrToReal8(Input2_pr,real_Input2,Input2_sz)
      call mxCopyPtrToReal8(Input3_pr,real_Input3,Input3_sz)
c      subroutine pr_equil_hydrocarbons(t,p,x,yy) ' t is also an output
      call pr_equil_hydrocarbons(real_Input1, real_Input2, real_Input3, real_Output2)
      real_Output1 = real_Input1
      call mxCopyReal8ToPtr(real_Output1,Output1_pr,1)
      call mxCopyReal8ToPtr(real_Output2,Output2_pr,nj)
c----- ! Call equilibrium function_polar

```

```

        elseif ((Func_name.eq.'equilibrium_polar') .and. (lib_Inited.eq.1)) then
            if (nrhs .ne. 4) then
                call mexErrMsgTxt('Three inputs (T, P, x) is required. - (equilibrium) [thermo_LIBRARY.dll]')
            elseif (nlhs .ne. 2) then
                call mexErrMsgTxt('Two outputs (T and vapor comp.) are required. - (equilibrium) [thermo_LIBRARY.dll]')
            endif
            if (mxIsNumeric(prhs(2)) .ne. 1) call mexErrMsgTxt('Input #1 is not a numeric. - (equilibrium) [thermo_LIBRARY.dll]')
            if (mxIsNumeric(prhs(3)) .ne. 1) call mexErrMsgTxt('Input #2 is not a numeric. - (equilibrium) [thermo_LIBRARY.dll]')
            if (mxIsNumeric(prhs(4)) .ne. 1) call mexErrMsgTxt('Input #3 is not a numeric. - (equilibrium) [thermo_LIBRARY.dll]')
            m = mxGetM(prhs(2))
            n = mxGetN(prhs(2))
            Input1_sz = m*n
            if (n .ne. 1 .or. m .ne. 1) call mexErrMsgTxt('Input #1 is not a scalar. - (equilibrium) [thermo_LIBRARY.dll]')
            m = mxGetM(prhs(3))
            n = mxGetN(prhs(3))
            Input2_sz = m*n
            if (n .ne. 1 .or. m .ne. 1) call mexErrMsgTxt('Input #2 is not a scalar. - (equilibrium) [thermo_LIBRARY.dll]')
            m = mxGetM(prhs(4))
            n = mxGetN(prhs(4))
            Input3_sz = m*n
            if (n .ne. nj .or. m .ne. 1)
& call mexErrMsgTxt('Input #3 is not a NC-element row vector. - (equilibrium) [thermo_LIBRARY.dll]')
                Input1_pr = mxGetPr(prhs(2))
                Input2_pr = mxGetPr(prhs(3))
                Input3_pr = mxGetPr(prhs(4))
                plhs(1) = mxCreateFull(1,1,0)
                plhs(2) = mxCreateFull(1,nj,0)
                Output1_pr = mxGetPr(plhs(1))
                Output2_pr = mxGetPr(plhs(2))
                status = 0
                allocate (real_Input1(1),STAT = alloc_err)
                status = status + alloc_err
                allocate (real_Input2(1),STAT = alloc_err)
                status = status + alloc_err
                allocate (real_Input3(nj),STAT = alloc_err)
                status = status + alloc_err
                allocate (real_Output1(1),STAT = alloc_err)
                status = status + alloc_err
                allocate (real_Output2(nj),STAT = alloc_err)
                status = status + alloc_err
                if (status .ne. 0) then
                    if (allocated(real_Input1)) deallocate(real_Input1)
                    if (allocated(real_Input2)) deallocate(real_Input2)
                    if (allocated(real_Input3)) deallocate(real_Input3)
                    if (allocated(real_Output1)) deallocate(real_Output1)
                    if (allocated(real_Output2)) deallocate(real_Output2)
                    call mexErrMsgTxt('Memory allocation error. - (equilibrium) [thermo_LIBRARY.dll]')
                endif
                call mxCopyPtrToReal8(Input1_pr,real_Input1,Input1_sz)
                call mxCopyPtrToReal8(Input2_pr,real_Input2,Input2_sz)
                call mxCopyPtrToReal8(Input3_pr,real_Input3,Input3_sz)
c          subroutine pr_equil_polar(t,p,x,yy) ' t is also an output
                call pr_equil_polar(real_Input1, real_Input2, real_Input3, real_Output2)
                real_Output1 = real_Input1
                call mxCopyReal8ToPtr(real_Output1,Output1_pr,1)
                call mxCopyReal8ToPtr(real_Output2,Output2_pr,nj)
c ----- ! No relevant function
                else
                call mexErrMsgTxt('Library is not initialized or No relevant function is requested. - [thermo_LIBRARY.dll]')
                endif
c ----- ! Memory deallocation
                if (allocated(real_Input1)) deallocate(real_Input1)

```

```

        if (allocated(real_Input2)) deallocate(real_Input2)
        if (allocated(real_Input3)) deallocate(real_Input3)
        if (allocated(real_Input4)) deallocate(real_Input4)
        if (allocated(real_Input5)) deallocate(real_Input5)
        if (allocated(real_Input6)) deallocate(real_Input6)
        if (allocated(int_Input1)) deallocate(int_Input1)
        if (allocated(int_Input2)) deallocate(int_Input2)
        if (allocated(int_Input3)) deallocate(int_Input3)
        if (allocated(int_Input4)) deallocate(int_Input4)
        if (allocated(int_Input5)) deallocate(int_Input5)
        if (allocated(int_Input6)) deallocate(int_Input6)
        if (allocated(real_Output1)) deallocate(real_Output1)
        if (allocated(real_Output2)) deallocate(real_Output2)
        if (allocated(real_Output3)) deallocate(real_Output3)
        if (allocated(real_Output4)) deallocate(real_Output4)
        if (allocated(real_Output5)) deallocate(real_Output5)
        if (allocated(real_Output6)) deallocate(real_Output6)
        if (allocated(int_Output1)) deallocate(int_Output1)
        if (allocated(int_Output2)) deallocate(int_Output2)
        if (allocated(int_Output3)) deallocate(int_Output3)
        if (allocated(int_Output4)) deallocate(int_Output4)
        if (allocated(int_Output5)) deallocate(int_Output5)
        if (allocated(int_Output6)) deallocate(int_Output6)

    return
end

C -----
C ----- thermo_LIBRARY Initialization routine
C -----
    subroutine init(check_input,st)
        integer check_input, st
        include 'thermo_LIBRARY.h'
        include 'parameter.h'
        include 'common_plant.h'
C ----- Initialization of the 'plant' common statement in 'common_plant.h' -----
C ----- written by MTD (Revised by Uğur Yıldız)
        integer :: i,j,I_O_err
        integer :: thermo_LIBRARY_dummy_pr, thermo_LIBRARY_dummy_pi
C -----
        tolerance = 1.d-7
        open(5,file='thermo_data.dat',IOSTAT=I_O_err, ERR = 100)
        read(5,*)
        read(5,*) tolerance
        read(5,*)
        read(5,*)
        do i=1,nj
            read(5,*) mw(i),tc(i),tboil(i),pc(i),wc(i)
        enddo
        read(5,*)
        read(5,*)
        do i=1,nj
            read(5,*) (del(i,j),j=1,nj)
        enddo
        read(5,*)
        read(5,*)
        do i=1,nj
            read(5,*) cenh1(i),cenh2(i),cenh3(i),cenh4(i)
        enddo
        close(5)
C -----
        if (check_input .eq. 1) then
            open(6,file='thermo_data_check.dat')
            write(6,*) 'tolerance'
            write(6,1) tolerance

```

```

write(6,*)
write(6,*) 'Mw(kg/mol)  Tc(K)  Tboil(K)  Pc(Pa)      w'
do i=1,nj
    write(6,2) mw(i),tc(i),tboil(i),pc(i),wc(i)
enddo
write(6,*)
write(6,*) 'del(binary interaction parameters)'
do i=1,nj
    write(6,3) (del(i,j),j=1,nj)
enddo
write(6,*)
write(6,*) ' cenh1    cenh2    cenh3    cenh4(J/molK)'
do i=1,nj
    write(6,4) cenh1(i),cenh2(i),cenh3(i),cenh4(i)
enddo
close(6)
endif
lib_inited = 1
st = 0
return
100 if (I_O_err.ne.0) then
    call mexErrMsgTxt("thermo_data.dat" couldn't be opened. - (init) [thermo_LIBRARY.dll])
    lib_inited = 0
    st = 1
    return
endif
1  format(d11.3)
2  format(5d15.3)
3  format(d13.1,;,d13.1,;,d13.1,;,d13.1,;,d13.1,;,d13.1,;,d13.1,;,d13.1,;,d13.1,;,d13.1,;,d13.1,;,d13.1,;,d13.1)
4  format(4d15.3)
end subroutine

C -----
C Write statements in these routines are exchanged with mexErrMsgTxt and mexPrintf
c and also 'parameter.h' and 'plant_data.dat' are modified.
C -----
C -----
C Peng-Rabinson EOS Subroutines Written by Mustafa T. DOKUCU
C Date : 16-05-2001
C -----
c -----

subroutine enth(t,p,x,y,hl,hv)
!Usage:
!       to calculate the ideal gas mixture enthalpy
!Record of revisions:
!      date          programmer          description of change
!      ====          =====          =====
! 18/03/2001  MTD          original code
!
implicit none
include 'parameter.h'
include 'common_plant.h'
!
! Inputs
!
! =====
real*8 :: t          !temperature
real*8 :: p          !pressure
real*8 :: x(nj)      !liquid phase fractions
real*8 :: y(nj)      !vapour phase fractions
!
! Locals
!
! =====
real*8 :: hl1        !ideal liquid mixture enthalpy
real*8 :: hv1        !ideal vapour mixture enthalpy
real*8 :: dhl        !liquid enthalpy departure

```

```

real*8 :: dhv          !vapour enthalpy departure
real*8 :: enigl       !ideal gas enthalpy
real*8 :: enigv       !ideal gas enthalpy
real*8 :: c1,cv1
real*8 :: c2,cv2
real*8 :: c3,cv3
real*8 :: c4,cv4
integer:: i
integer:: ifase
!           Outputs
!           =====
real*8 :: hl          !liquid enthalpy
real*8 :: hv          !vapour enthalpy
c1 = 0.d0
cv1 = 0.d0
c2 = 0.d0
cv2 = 0.d0
c3 = 0.d0
cv3 = 0.d0
c4 = 0.d0
cv4 = 0.d0
do i = 1,nj
  c1 = c1 + cenh1(i) * x(i)
c2 = c2 + cenh2(i) * x(i)
c3 = c3 + cenh3(i) * x(i)
c4 = c4 + cenh4(i) * x(i)
  cv1 = cv1 + cenh1(i) * y(i)
cv2 = cv2 + cenh2(i) * y(i)
  cv3 = cv3 + cenh3(i) * y(i)
cv4 = cv4 + cenh4(i) * y(i)
enddo
enigl = c1 * (t-trf) +
& + (1.d0/2.d0) * c2 * (t**2 - trf**2)
& + (1.d0/3.d0) * c3 * (t**3 - trf**3)
& + (1.d0/4.d0) * c4 * (t**4 - trf**4)
  enigv = cv1 * (t-trf) +
& + (1.d0/2.d0) * cv2 * (t**2 - trf**2)
& + (1.d0/3.d0) * cv3 * (t**3 - trf**3)
& + (1.d0/4.d0) * cv4 * (t**4 - trf**4)
ifase = 0
  call pr_enth(t,p,x,ifase,dhl)
  ifase = 1
  call pr_enth(t,p,y,ifase,dhv)
  hl = enigl + dhl + 20000.d0
  hv = enigv + dhv + 20000.d0
return
end subroutine
c -----
subroutine pr_compr(a_mixture,b_mixture,z_liq,z_vap)
!Usage:
!           to solve the cubic equation for the liquid and vapor
!compressibility factors used for the estimation of species
!fugacities
!Record of revisions:
!           date           programmer           description of change
!           =====
! 14/02/2001  MTD           original code
implicit none
include 'parameter.h'
include 'common_plant.h'
!           Inputs
!           =====

```

```

real*8 :: a_mixture
real*8 :: b_mixture
!           Locals
!           =====
complex*8:: z_vap_cplx
complex*8:: z_liq_cplx
complex*8:: s1
complex*8:: a
complex*8:: b
!           Outputs
!           =====
real*8:: z_vap
real*8:: z_liq
!convert the type declaration of the input variables to complex
a = cmplx(a_mixture,0.d0)
b = cmplx(b_mixture,0.d0)
!calculate the liquid phase compressibility
s1 = (-36.0d0*a+144.0d0*a*b-48.0d0*b**2-224.0d0*b**3+48.0d0*b+8.0d0+12.0d0*sqrt(24.0d0*a*b-24.0d0*b**
&2-192.0d0*b**3+264.0d0*a*b**2-3.0d0*a**2+24.0d0*a**2*b**2-120.0d0*a**2*b-48.0d0*a*b**4+336.0d0
&a*b**3-480.0d0*b**4+12.0d0*a**3-96.0d0*b**6-384.0d0*b**5))*(1.d0/3.d0)/12.0d0+(a-10.
&d0/3.d0*b**2-4.d0/3.d0*B-1.d0/3.d0)/(-36.0d0*a+144.0d0*a*b-48.0d0*b**2-224.0d0*b**
&3.0d0+48.0d0*b+8.0d0+12.0d0*sqrt(24.0d0*a*b-24.0d0*b**2-192.0d0*b**3+264.0d0*a*b**2-3.0d0*a**2+24.0d0*a**2
&b**2-120.0d0*a**2*b-48.0d0*a*b**4+336.0d0*a*b**3-480.0d0*b**4+12.0d0*a**3-96.0d0*b**6-384.0d0
&*B**5))*(1.d0/3.d0)
z_liq_cplx = s1+1.d0/3.d0-b/3.d0+cmplx(0.d0,1.d0)*sqrt(3.d0)*((-36.d0*a+144.d0*a*b-
&48.d0*b**2-224.d0*b**3+48.d0*b+8.0d0+12*sqrt(24.d0*a*b-24.0d0*b**2-192.d0*b**3+264.d0*a*b**2
&-3.d0*a**2+24.d0*a**2*b**2-120.d0*a**2*b-48.0d0*a*b**4+336.d0*a*b**3-480.d0*b**4+12.d0*a
&**3-96.d0*b**6-384.d0*b**5))*(1.d0/3.d0)/6.d0+(2.d0*a-20.d0/3.d0*b**2-8.d0/3.
&d0*b-2.d0/3.d0)/(-36.d0*a+144.d0*a*b-48.d0*b**2-224.d0*b**3+48.d0*b+8.0d0+12.d0*sqrt(24.d0*
&a*b-24.0d0*b**2-192.d0*b**3+264.d0*a*b**2-3.0d0*a**2+24.0d0*a**2*b**2-120.d0*a**2*b-48.d0*
&a*b**4+336.d0*a*b**3-480.d0*b**4+12.d0*a**3-96.d0*b**6-384.d0*b**5))*(1.d0/3.d0)
&)/2.d0
!calculate the vapor phase compressibility
z_vap_cplx =(-36.d0*a+144.d0*a*b-48.d0*b**2-224.d0*b**3+48.d0*b+8.0d0+12.d0*sqrt(24.d0*a*b-24.d0*b**2
&-192.d0*b**3+264.d0*a*b**2-3.0d0*a**2+24.0d0*a**2*b**2-120.d0*a**2*b-48.d0*a*b**4+336.d0*
&a*b**3-480.d0*b**4+12.d0*a**3-96.d0*b**6-384.d0*b**5))*(1.d0/3.d0)/6.d0-(2.0d0*a-20.
&d0/3.d0*b**2-8.d0/3.d0*b-2.d0/3.d0)/(-36.d0*a+144.d0*a*b-48.d0*b**2-224.d0*b**
&3+48.d0*b+8.0d0+12.d0*sqrt(24.d0*a*b-24.0d0*b**2-192.d0*b**3+264.d0*a*b**2-3.0d0*a**2+24.0d0*a**2
&b**2-120.d0*a**2*b-48.d0*a*b**4+336.d0*a*b**3-480.d0*b**4+12.d0*a**3-96.d0*b**6-384.d0
&*b**5))*(1.d0/3.d0)+1.d0/3.d0-b/3.d0
!there is no liquid phase if the liquid compressibility root is a complex #
!in this case the compressibility root returned as equal to vapor phase
!compressibility
if (aimag(z_liq_cplx) > tolerance) then
z_liq = real(z_vap_cplx)
else
z_liq = real(z_liq_cplx)
endif
!the root found for the vapor compressibility is erroneous if it is
!a complex # in this case the compressibility root returned as zero to
!the mainprogram
if (aimag(z_vap_cplx) > tolerance) then
call mexPrintf('vapor phase compressibility can not be calculated. - (pr_compr) [thermo_LIBRARY.dll]\n')
c write(*,*) 'vapor phase compressibility can not be calculated'
z_vap = 0.d0
else
z_vap = real(z_vap_cplx)
endif
return
end subroutine
c -----
subroutine pr_cons(t,a,aij,b)

```



```

!Usage:
!           to calculate the constants A and B of the
!Peng-Robinson EOS which is explained in p239 (Sandler)
!Record of revisions:
!           date           programmer           description of change
!           ====           =====           =====
! 12/03/2001  MTD           original code
implicit none
include 'parameter.h'
include 'common_plant.h'
!           Inputs
!           =====
real*8 :: t           !temperature [K]
!
!           Locals
!           =====
real*8 :: ac(nj)           !constant
real*8 :: xk           !constant
real*8 :: alsqr           !constant
real*8 :: alpha           !constant
real*8 :: tr           !reduced temperature
integer :: i
integer :: j
!           Outputs
!           =====
real*8 :: a(nj)           !a of the species
real*8 :: b(nj)           !b of the species
real*8 :: aij(nj,nj)           !interacting a's of the species
do i=1,nj
ac(i) = 0.457235529d0 * ((rg * tc(i))**2) / pc(i)           !eqn 4.7-1(first part)
b(i) = 7.779607400000001d-2 * rg * tc(i) / pc(i)           !eqn 4.7-2
xk = 0.37464d0 + (1.54226d0 - 0.26992d0 * wc(i)) * wc(i) !eqn 4.7-4
tr = t / tc(i)
alsqr = 1.d0 + xk * (1.d0 - dsqrt(tr))
alpha = alsqr * alsqr           !eqn 4.7-3
a(i) = alpha * ac(i)           !eqn 4.7-1(whole)
enddo
do i=1,(nj-1)
do j=(i+1),nj
aij(i,j) = (1.d0 - del(i,j)) * dsqrt(a(i) * a(j))           !eqn 7.4-9
aij(j,i) = aij(i,j)
enddo
enddo
return
end subroutine

```

---

```

subroutine pr_dens_l(t,p,x,mwa,densa)
!Usage:
!           to calculate the average molecular weight and the
!density of the liquid phase using Peng-Robinson EOS
!Record of revisions:
!           date           programmer           description of change
!           ====           =====           =====
! 25/03/2001  MTD           original code
! 20/11/2006  HC           modification to liquid density
implicit none
include 'parameter.h'
include 'common_plant.h'
!           Inputs
!           =====
real*8 :: t
real*8 :: p

```

```

        real*8 :: x(nj)
        !           Locals
        !           =====
        real*8 :: aa
        real*8 :: bb
real*8 :: ca
        real*8 :: cb
real*8 :: a(nj)
        real*8 :: b(nj)
        real*8 :: zx(nj)
real*8 :: aij(nj,nj)
        real*8 :: z_liq
        real*8 :: z_vap
        real*8 :: zz
        real*8 :: vv
        real*8 :: sumx
integer:: i
        integer:: j
        !           Outputs
        !           =====
        real*8 :: mwa
        real*8 :: densa
        mwa = 0.d0
        sumx = 0.d0
do i = 1,nj
    sumx = sumx + x(i)
    enddo
    do i = 1,nj
        zx(i) = x(i) / sumx
    enddo

    do i = 1,nj
mwa = mwa + mw(i) * zx(i)
    enddo
    call pr_cons(t,a,aij,b)
    aa = 0.d0
    bb = 0.d0
    do i = 1,nj
bb = bb + zx(i) * b(i)
        do j = 1,nj
            if (i == j) then
                aa = aa + zx(i) * zx(i) * a(i)
            else
                aa = aa + zx(i) * zx(j) * aij(i,j)
            endif
        enddo
    enddo
ca = aa * p / ((rg * t)**2)
    cb = bb * p / (rg * t)
    call pr_compr(ca,cb,z_liq,z_vap)
    zz = z_liq
    vv = zz * rg * t / p
    densa = mwa / vv
    return
end subroutine
c -----
subroutine pr_dens_g(t,p,x,mwa,densa)
!Usage:
!           to calculate the average molecular weight and the
!density of the vapor phase using Peng-Robinson EOS
!Record of revisions:
!           date           programmer           description of change

```

```

! =====
! 25/03/2001 MTD original code
! 20/11/2006 HC modification to vapor density
implicit none
include 'parameter.h'
include 'common_plant.h'
! Inputs
! =====
real*8 :: t
real*8 :: p
real*8 :: x(nj)
! Locals
! =====
real*8 :: aa
real*8 :: bb
real*8 :: ca
real*8 :: cb
real*8 :: a(nj)
real*8 :: b(nj)
real*8 :: zx(nj)
real*8 :: aij(nj,nj)
real*8 :: z_liq
real*8 :: z_vap
real*8 :: zz
real*8 :: vv
real*8 :: sumx
integer:: i
integer:: j
! Outputs
! =====
real*8 :: mwa
real*8 :: densa
mwa = 0.d0
sumx = 0.d0
do i = 1,nj
sumx = sumx + x(i)
enddo
do i = 1,nj
zx(i) = x(i) / sumx
enddo
do i = 1,nj
mwa = mwa + mw(i) * zx(i)
enddo
call pr_cons(t,a,aij,b)
aa = 0.d0
bb = 0.d0
do i = 1,nj
bb = bb + zx(i) * b(i)
do j = 1,nj
if (i == j) then
aa = aa + zx(i) * zx(i) * a(i)
else
aa = aa + zx(i) * zx(j) * aij(i,j)
endif
endif
enddo
enddo
ca = aa * p / ((rg * t)**2)
cb = bb * p / (rg * t)
call pr_compr(ca,cb,z_liq,z_vap)
zz = z_vap
vv = zz * rg * t / p
densa = mwa / vv

```

```

return
end subroutine
C -----
subroutine pr_enth(t,p,zx,ifase,dh)
!Usage:
!to calculate the enthalpy departure of a mixture
!as explained in Sandler p425
!Peng-Robinson EOS is explained in p239
!Record of revisions:
! date           programmer      description of change
!  =====
!12/03/2001      MTD             original code
implicit none
include 'parameter.h'
include 'common_plant.h'
!           Inputs
!           =====
real*8 :: t
real*8 :: p
real*8 :: zx(nj)
!           Locals
!           =====
real*8 :: zz
real*8 :: a(nj)
real*8 :: b(nj)
real*8 :: xk(nj)
real*8 :: aij(nj,nj)
real*8 :: c1,c2,c3,c4,c5,c6,c7,c8,c9
real*8 :: a1,a2,a3,a4
real*8 :: anum,aden
real*8 :: cnum,cden
real*8 :: damdt
real*8 :: aa
real*8 :: ca
real*8 :: bb
real*8 :: cb
real*8 :: z_liq
real*8 :: z_vap
real*8 :: tr(nj)
real*8 :: dh0,dh1,dh2
integer:: i
integer:: j
integer:: ifase
!           Outputs
!           =====
real*8:: dh
call pr_cons(t,a,aij,b)
aa = 0.d0
bb = 0.d0
do i = 1,nj
bb = bb + zx(i) * b(i)
do j = 1,nj
if (i == j) then
aa = aa + zx(i) * zx(i) * a(i)
else
aa = aa + zx(i) * zx(j) * aij(i,j)
endif
enddo
enddo
ca = aa * p / ((rg * t)**2)
cb = bb * p / (rg * t)
call pr_compr(ca,cb,z_liq,z_vap)

```

```

        if (ifase == 0) then
            zz = z_liq
        else
            zz = z_vap
        endif
        do i = 1,nj
            xk(i) = 0.37464d0 + (1.54226d0 - 0.26992d0 * wc(i)) * wc(i) !eqn 4.7-4
        enddo
        damdt = 0.d0
    do i = 1,nj
        do j = 1,nj
            tr(i) = t / tc(i)
            tr(j) = t / tc(j)
            c1 = (-0.457235529d0/2.d0) * (-1 + del(i,j)) * rg**4
            c2 = tc(i) * (-1.d0 - xk(i) + xk(i) * dsqrt(tr(i)))
            c3 = tc(j) * (-1.d0 - xk(j) + xk(j) * dsqrt(tr(j)))
            c4 = -tc(j) * xk(i) * dsqrt(tr(j))
            & -tc(j) * xk(i) * dsqrt(tr(j)) * xk(j)
            c5 = 2.d0 * xk(i) * t * xk(j)
            & -tc(i) * xk(j) * dsqrt(tr(i))
            c6 = -tc(i) * xk(j) * dsqrt(tr(i)) * xk(i)
                c7 = (-1.d0 - xk(i) + xk(i) * dsqrt(tr(i)))**2
                c8 = (-1.d0 - xk(j) + xk(j) * dsqrt(tr(j)))**2 / pc(i) / pc(j)
            c9 = pc(i) * pc(j) * dsqrt(tr(i)) * dsqrt(tr(j))
            cnum = c4 + c5 + c6
                cden = dsqrt(rg**4 * tc(i)**2 * c7 * tc(j)**2 * c8)
            & * pc(i) * pc(j) * dsqrt(tr(i)) * dsqrt(tr(j))
            damdt = damdt + zc(i) * zc(j) * (c1 * c2 * c3 * cnum / cden)
        enddo
    enddo
    a1 = dsqrt(2.d0)
    a2 = a1 + 1.d0
    a3 = a1 - 1.d0
    a4 = a1 * 2.d0
    anum = zz + (a2 * cb)
    aden = zz - (a3 * cb)
    dh0 = (rg * t)*(zz - 1.d0)
    dh1 = (t*damdt - aa)/a4/bb
    dh2 = dlog(anum/aden)
    dh = (dh0 + dh1*dh2)
return
end subroutine
c -----
subroutine pr_equil_hydrocarbons(t,p,x,yy)
!Usage:
!      to calculate the bubble point temperature using
!Peng-Robinson EOS similar to VLMU.BAS of Sandler
!Record of revisions:
!      date          programmer          description of change
!      =====
! 12/03/2001  MTD          original code
! 12/06/2001  UGUR        to be able to find equilibrium staff
!                        when a zero-fraction component exist.
!                        some checks were performed before calculation.
!
implicit none
include 'parameter.h'
include 'common_plant.h'
!      Inputs
!      =====
real*8:: t          ! t is also an output
real*8:: p
real*8:: x(nj)

```

```

!           Locals
!           =====
real*8 :: s(2),sum,sumy
real*8 :: dt1,dt2
real*8 :: dlnp
real*8 :: ps(nj)
real*8 :: xk1(nj)
real*8 :: a(nj)
real*8 :: aij(nj,nj)
real*8 :: b(nj)
real*8 :: zx(nj)
real*8 :: fugacity(nj)
real*8 :: f1(nj),f2(nj)
real*8 :: zz,zz1,zz2
real*8 :: y1(nj),y2(nj)
real*8 :: yk
real*8 :: test,ttest
real*8 :: dsdt,dlt,dd
real*8 :: neg_dd,neg_dlt
real*8 :: tbg,tcg
integer:: i
integer:: j
integer:: k
integer:: nc
integer:: kkk
integer:: nloop
integer:: iconv
integer:: ifase
integer:: itest
integer:: kvalue
logical:: reguess
!           Outputs
!           =====
real*8 yy(nj)
integer:: comp_index(nj)
common /nc/ nc
! zero component check
j=0
do i=1,nj
    if (x(i).gt.0.0d0) then
        j=j+1
        comp_index(j) = i
    else
        yy(i) = 0.0d0
    endif
end do
nc = j
k   = 0
kvalue = 0
iconv = 0
sum   = 0.d0
reguess = .false.

do i = 1,nc
    sum = sum + x(comp_index(i))
enddo
do i = 1,nc
    x(comp_index(i)) = x(comp_index(i))/sum
enddo
!Initial guess procedure for Ki = yi / xi and yi
if (k > 30) then
    call mexErrMsgTxt("too many calculations. - (pr_equal) [thermo_LIBRARY.dll]")

```

3870

```

c   write(*,*) 'too many calculations'
      goto 4980
    endif
    do i = 1,nc
      if (kvalue == 1) then
yy(comp_index(i)) = x(comp_index(i)) * xk1(comp_index(i))
      else
        dt1 = (1.d0 / t) - (1.d0 / tboil(comp_index(i)))
        dt2 = (1.d0 / tc(comp_index(i))) - (1.d0 / tboil(comp_index(i)))
        dlnp = dlog(pc(comp_index(i)))
        ps(comp_index(i)) = dexp(dlnp * dt1 / dt2)
        xk1(comp_index(i)) = ps(comp_index(i)) / p
        yy(comp_index(i)) = x(comp_index(i)) * xk1(comp_index(i))
      endif
    enddo

!Calculation of liquid fugacities
kkk = 0
3980 nloop = 1
      if (t < 50.d0) goto 4590
      if (t > 1200.d0) goto 4590
      k = k + 1
      !call pr_cons
4020 call pr_cons(t,a,aij,b)
      do i=1,nc
        zx(comp_index(i)) = x(comp_index(i))
      enddo
!phase liquid 1
ifase = 1
!call pr_fuga
{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}
call pr_fuga(t,p,ifase,zx,zz,a,aij,b,fugacity)
!{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}
GUR}{UGUR}
      do i = 1,nc
        f1(comp_index(i)) = fugacity(comp_index(i))
      enddo
      zz1 = zz
      sumy = 0.d0
      do i = 1,nc
        y2(comp_index(i)) = yy(comp_index(i))
        sumy = sumy + yy(comp_index(i))
      enddo
      do i = 1,nc
        yy(comp_index(i)) = yy(comp_index(i)) / sumy
      enddo
4160 sumy = 0.d0
4170 kkk = kkk + 1
      do i = 1,nc
        zx(comp_index(i)) = yy(comp_index(i))
      enddo
!phase vapor 0
ifase = 0
!call pr_fuga
{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}
call pr_fuga(t,p,ifase,zx,zz,a,aij,b,fugacity)
!{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}{UGUR}
GUR}{UGUR}
      do i = 1,nc
        f2(comp_index(i)) = fugacity(comp_index(i))
      enddo
      zz2 = zz
      do i = 1,nc

```

```

yy(comp_index(i)) = f1(comp_index(i)) * yy(comp_index(i)) / f2(comp_index(i))
y1(comp_index(i)) = yy(comp_index(i))
sumy = sumy + yy(comp_index(i))
enddo
itest = 0
do i = 1,nc
  test = dabs(y1(comp_index(i)) - y2(comp_index(i)))
  if (test > tolerance) then
    itest = itest + 1
  endif
  yy(comp_index(i)) = yy(comp_index(i)) / sumy
enddo
if (kkk > 25) goto 4400
if (itest <= 0) goto 4400
do i = 1,nc
  y2(comp_index(i)) = y1(comp_index(i))
enddo
goto 4160
4400 s(nloop) = sumy
kkk = 0
if ((nloop - 1) > 0) goto 4460
nloop = 2
t = t - 0.005d0
goto 4020
4460 dsdt = (s(2) - s(1)) / (0.005d0)
if (dabs(dsdt) < tolerance) goto 4590
dlt = (s(1) - 1.d0) / dsdt
if (dabs(dlt) < 0.0026) goto 4670
if (k > 50) goto 4650
if (k < 11) then
  dd = 20.d0
endif
if (k >= 11) then
  dd = 5.d0
endif
if (dlt > dd) then
  t = t + dd
endif
if (dlt > dd) goto 3980
neg_dd = -1.d0 * dd
if (dlt < neg_dd) then
  t = t - dd
endif
neg_dlt = -1.d0 * dlt
if (neg_dlt > dd) goto 3980
t = t + dlt + 0.0025
goto 3980
4590 if (k > 2) goto 4630
if (zz1 >= 0.307d0) then
  t = t - 10.d0
endif
if (zz1 <= 0.307d0) then
  t = t + 10.d0
endif
goto 3980
4630 call mexErrMsgTxt('not converging: one-phase region or poor initial guess. - (pr_equil) [thermo_LIBRARY.dll]')
c4630 write(*,*) 'not converging: one-phase region or poor initial guess'
if (reguess == .false.) then
  reguess = .true.
enddo
tbg = 0.d0
tcg = 0.d0
do i = 1,nc

```



```

        tbg = tbg + x(comp_index(i)) * tboil(comp_index(i))
        tcg = tcg + x(comp_index(i)) * tc(comp_index(i))
    enddo

    t = 0.5d0 * (tbg + tcg)
    call mexPrintf('it should print "t" here. - (pr_equil) [thermo_LIBRARY.dll]')
c
    write(*,*) t
    goto 3870
    else
        goto 4880
    endif
4650
    call mexErrMsgTxt('bubble point temperature calculation did not converge. - (pr_equil) [thermo_LIBRARY.dll]')
c4650
    write(*,*) 'bubble point temperature calculation did not converge'
    goto 4880
4670
    yk = k
    ttest = (zz1 - zz2)**2
    if (ttest > tolerance) goto 4730
    if (zz1 >= 0.307d0) then
        t = t - 25.d0 / dsqrt(yk)
    endif
    if (zz1 < 0.307d0) then
        t = t + 25.d0 / dsqrt(yk)
    endif

    kvalue = 0
    goto 3870
4730
    do i = 1,nc
        xk1(comp_index(i)) = yy(comp_index(i)) / x(comp_index(i))
    enddo
    return
4880
    call mexErrMsgTxt('unsuccessful bubble point calculation. - (pr_equil) [thermo_LIBRARY.dll]')
c4880
    write(*,*) 'unsuccessful bubble point calculation'
    end subroutine
c -----
subroutine pr_equil_polar(t,p,x,yy)
!Usage:
!
!       to calculate the bubble point temperature using
!Peng-Robinson EOS similar to VLMU.BAS of Sandler
!Record of revisions:
!       date           programmer           description of change
! =====
! 12/03/2001 MTD
! 12/06/2001 UGUR
!
! original code
! to be able to find equilibrium staff
! when a zero-fraction component exist.
! some checks were performed before calculation.
!
! ALMILA
! gama-fi approach using NRTL activity coeff. model
! 05/03/2007 HATICE
! modification of calculation of vapor pressure and
! predictions of binary interaction parameters for ethanol and water
implicit none
include 'parameter.h'
include 'common_plant.h'
!
!       Inputs
!       =====
real*8 :: t           ! t is also an output
real*8 :: p
real*8 :: x(nj)
!
!       Locals
!       =====
real*8 :: s(2),sum,sumy
real*8 :: dt1,dt2
real*8 :: dlnp
real*8 :: ps(nj)
real*8 :: xk1(nj)
real*8 :: a(nj)

```

```

real*8 :: aij(nj,nj)
real*8 :: b(nj)
real*8 :: zx(nj)
real*8 :: fugacity(nj)
real*8 :: f1(nj),f2(nj)
real*8 :: zz,zz1,zz2
real*8 :: y1(nj),y2(nj)
real*8 :: yk
real*8 :: test,ttest
real*8 :: dsdt,dlt,dd
real*8 :: neg_dd,neg_dlt
real*8 :: tbg,tcg
integer:: i
integer:: j
integer:: k
integer:: nc
integer:: kkk
integer:: nloop
integer:: iconv
integer:: ifase
integer:: itest
integer:: kvalue
logical:: reguess
real*8 :: ant_cons_a(nj),ant_cons_b(nj),ant_cons_c(nj), ant_cons_d(nj)
real*8 :: gama(nj),sumy1
integer:: loop
real*8 :: alact(nj),tagx1,tagx2,total,xgkj,xgij1
real*8 :: alp(nj,nj),ta(nj,nj),gij(nj,nj),x_pr(nj)
real*8 :: rg_cal
!           Outputs
!           =====
real*8 yy(nj)
integer:: comp_index(nj)
common /nc/ nc
! zero component check
j=0
do i=1,nj
    if (x(i).gt.0.0d0) then
        j=j+1
        comp_index(j) = i
    else
        yy(i) = 0.0d0
    endif
enddo
nc = j
c (1) ethanol (2) water !Wagner equation constants
ant_cons_a(1) = -8.51838d0
ant_cons_a(2) = -7.76451d0
ant_cons_b(1) = 0.34163d0
ant_cons_b(2) = 1.45838d0
ant_cons_c(1) = -5.73683d0
ant_cons_c(2) = -2.77580d0
ant_cons_d(1) = 8.32581d0
ant_cons_d(2) = -1.23303d0
sum = 0.0d0
do i = 1,nc
    sum = sum + x(comp_index(i))
enddo
do i = 1,nc
    x(comp_index(i)) = x(comp_index(i))/sum
enddo
loop = 0

```

```

10      loop = loop+1
        if (loop .gt. 1800) go to 4630
        sumy = 0.0d0
c       Activity coefficients calculation from NRTL model
c       (1) ethanol (2) water
        aij(1,2) = -175.0164d0
        aij(2,1) = 1440.3479d0
        alp(1,2) = 0.2959d0
        do i=1,nj-1
          do j=i+1,nj
            alp(j,i) = alp(i,j)
          enddo
        enddo
c       do i=1,nj
c       ta(i,i) = 0.0d0
c       enddo
c       rg_cal = rg/4.184 !gas constant [cal/(mol.K)]
        do i=1,nj
          do j=1,nj
            if (i .eq. j) then
              ta(i,j) = 0.0d0
            else
              ta(i,j) = aij(i,j) / (rg_cal * t)
            endif
            gij(i,j) = dexp(-alp(i,j)*ta(i,j))
          enddo
        enddo
        if (nc .eq. 2) then
          do i=1,nc
            do j=1,nc
              if (i .eq. j) then
                else
                  gama(comp_index(i)) = x(comp_index(j))**2*(ta(j,i)
&      *(gij(comp_index(j),comp_index(i))/(x(comp_index(i)
&      +x(comp_index(j))*gij(comp_index(j),comp_index(i))))**2
&      + ta(i,j)* gij(comp_index(i),comp_index(j))/ (x(comp_index(j))
&      + x(comp_index(i)) * gij(comp_index(i),comp_index(j))))**2)
                  gama(comp_index(i)) = dexp(gama(comp_index(i)))
                endif
              enddo
            enddo
          else
            do i=1,nc
              tagx1 = 0.0d0
              xgij1 = 0.0d0
              do j = 1,nc
                tagx1 = tagx1 + ta(comp_index(j),comp_index(i))*gij(comp_index(j),comp_index(i))*x(comp_index(j))
                xgij1 = xgij1 + x(comp_index(j))*gij(comp_index(j),comp_index(i))
              enddo
              total = 0.0d0
              do j=1,nc
                xgkj = 0.0d0
                do k=1,nc
                  xgkj = xgkj + x(comp_index(k))*gij(comp_index(k),comp_index(j))
                enddo
                tagx2 = 0.0d0
                do k=1,nc
                  tagx2 = tagx2 + x(comp_index(k))*ta(comp_index(k),comp_index(j))*gij(comp_index(k),comp_index(j))
                enddo
                total = total + x(comp_index(j))*gij(comp_index(i),comp_index(j))/xgkj*
&(ta(comp_index(i),comp_index(j)) - tagx2/xgkj)
              enddo
            enddo
          enddo
        enddo

```

```

        alact(comp_index(i)) = tagx1 / xgij1 + total
        gama(comp_index(i)) = dexp(alact(comp_index(i)))
        enddo

        endif
c      End of activity coefficients calculation
!     Vapor Pressure calculation
      do i=1,nc
        x_pr(comp_index(i))= (1 - t / tc(comp_index(i)))
          ps(comp_index(i))= (1-x_pr(comp_index(i)))**(-1)
        & *((ant_cons_a(comp_index(i)) * x_pr(comp_index(i)))
        & + ( ant_cons_b(comp_index(i))* x_pr(comp_index(i))**(15/10) )
        & +( ant_cons_c(comp_index(i))* x_pr(comp_index(i))**3 )
        & +( ant_cons_d(comp_index(i))* x_pr(comp_index(i))**6 ))
          ps(comp_index(i)) = dexp(ps(comp_index(i)))
        ps(comp_index(i))= ps(comp_index(i))*pc(comp_index(i))
        enddo
!     End of vapor pressure calculation
      do i=1,nc
        yy(comp_index(i)) = x(comp_index(i))*gama(comp_index(i))*ps(comp_index(i))/p
        sumy = sumy + yy(comp_index(i))
        enddo
      do j=1,nc
        y2(comp_index(j)) = yy(comp_index(j))*1.0d0/sumy
        yy(comp_index(j)) = y2(comp_index(j))
        enddo
!call pr_cons
      call pr_cons(t,a,aij,b)
11     do i = 1,nc
        zx(comp_index(i)) = yy(comp_index(i))
        enddo
!phase vapor 0
      ifase = 0
!call pr_fuga
      call pr_fuga(t,p,ifase,zx,zz,a,aij,b,fugacity)
      do i = 1,nc
        f2(comp_index(i)) = fugacity(comp_index(i))
        enddo
      sumy1 = 0.0d0
      do i=1,nc
        !y*p*fiv=x*gama*pvap
        yy(comp_index(i)) = x(comp_index(i))*gama(comp_index(i))*ps(comp_index(i))*yy(comp_index(i))/f2(comp_index(i))
        if (yy(comp_index(i)) .lt. 1.0d-16) yy(comp_index(i))=0.0d0
        if (yy(comp_index(i)) .gt. 1.0d0) yy(comp_index(i))=1.0d0
        enddo
      do i=1,nc
        sumy1 = sumy1+yy(comp_index(i))
        enddo
      dsdt = (sumy-sumy1)/0.005d0
      if ((sumy-sumy1) .lt. tolerance) go to 12
      sumy=sumy1
      do i=1,nc
        yy(comp_index(i)) = yy(comp_index(i))*1.0d0/sumy1
        enddo
      go to 11
12     if (dabs((sumy1-1.0d0)/dsdt) < 0.0026d0) return
      dlt = (sumy1-1.0d0)/dsdt
      dd = 5.0d0
      if (dlt > dd) then
        t = t+dd
      endif
      if (dlt > dd) go to 10

```

```

neg_dd = -1.d0 * dd
if (dlt < neg_dd) then
t = t-dd
endif
neg_dlt = -1.0d0 * dlt
if (neg_dlt > dd) go to 10
t = t + dlt + 0.0025d0
go to 10
4630 call mexErrMsgTxt('not converging: one-phase region or poor initial guess. - (pr_equil) [thermo_LIBRARY.dll]')
return
end subroutine
c -----
subroutine pr_fuga(t,p,ifase,zx,zz,a,aij,b,fugacity)
!Usage:
!to calculate the species fugacity f(T,P,xi)
!as explained in Sandler p409
!Peng-Robinson EOS is explained in p239
!Record of revisions:
! date programmer description of change
! =====
!12/03/2001 MTD original code
! 12/06/2001 UGUR to be able to find equilibrium stuff
! when a zero-fraction component exist.
! some checks were performed before calculation.
!
implicit none
include 'parameter.h'
include 'common_plant.h'
! Inputs
! =====
real*8 :: t
real*8 :: p
real*8 :: zx(nj)
real*8 :: a(nj)
real*8 :: b(nj)
real*8 :: aij(nj,nj)
integer:: ifase
! Locals
! =====
real*8 :: c1
real*8 :: c2
real*8 :: c3
real*8 :: sa(nj)
real*8 :: aa
real*8 :: bb
real*8 :: cb
real*8 :: ca
real*8 :: zz
real*8 :: z_liq
real*8 :: z_vap
real*8 :: fox(nj)
real*8 :: ag1
real*8 :: ag2
real*8 :: ag3
integer:: nc
integer:: i
integer:: j
! Outputs
! =====
real*8 :: fugacity(nj)
integer:: comp_index(nj)
common /nc/ nc
! zero component check

```

```

j=0
do i=1,nj
    if (zx(i) .gt. 0.d0) then
        j=j+1
        comp_index(j) = i
    endif
end do
nc = j
c1 = dsqrt(2.d0)
c2 = 1.d0 + c1
c3 = c1 - 1.d0
do i = 1,nc
    sa(comp_index(i)) = 0.d0
enddo
aa = 0.d0
bb = 0.d0
do i = 1,nc
    bb = bb + zx(comp_index(i)) * b(comp_index(i))
    do j = 1,nc
        if (i == j) then
            aa = aa + zx(comp_index(i)) * zx(comp_index(i)) * a(comp_index(i))
            sa(comp_index(j)) = sa(comp_index(j)) + zx(comp_index(j)) * a(comp_index(j))
        else
            aa = aa + zx(comp_index(i)) * zx(comp_index(j)) * aij(comp_index(i),comp_index(j))
            sa(comp_index(j)) = sa(comp_index(j)) + zx(comp_index(i)) * aij(comp_index(i),comp_index(j))
        endif
    enddo
enddo
ca = aa * p / ((rg*t)**2)
cb = bb * p / (rg*t)
call pr_compr(ca,cb,z_liq,z_vap)
if (ifase == 0) then
    zz = z_vap
else
    zz = z_liq
endif
ag1 = (zz + c2 * cb) / (zz - c3 * cb)
ag1 = dlog(ag1)
ag2 = ca / (2.d0 * cb * c1)
do i = 1,nc
    ag3 = (2.d0 * sa(comp_index(i)) / aa) - (b(comp_index(i)) / bb)
    fox(comp_index(i)) = (b(comp_index(i)) * (zz-1.d0) / bb) - dlog(zz - cb) - ag1 * ag2 * ag3
    fox(comp_index(i)) = dexp(fox(comp_index(i)))
    fugacity(comp_index(i)) = zx(comp_index(i)) * p * fox(comp_index(i))
enddo
return
end subroutine

```

### C.3.8 thermo\_LIBRARY.h

```

common /thermo_LIBRARY/      lib_Inited
integer :: lib_Inited      ! Toggle for checking whether thermo_LIBRARY.dll is initialized.

```

### C.3.9 common\_plant.h

```

common /plant/
& whs,whr,ds,dr,wls,wlr,mvb,mvd,tolerance,
& mw(nj),tc(nj),tboil(nj),pc(nj),wc(nj),del(nj,nj),
& cenh1(nj),cenh2(nj),cenh3(nj),cenh4(nj)
real*8 :: whs,whr,ds,dr,wls,wlr,mvb,mvd,tolerance

```

```

real*8 :: mw,tc,tboil,pc,wc,dcl
real*8 :: cenh1,ceh2,ceh3,ceh4

```

### C.3.10 parameter.h

```

C ----- This parameters were modified as the common statement labeled as 'parameter'
C       integer,parameter :: nt = 100           ! number of trays
C
C       integer,parameter :: nj = 3             ! number of components
C -----

integer ,parameter :: nj = 3                   ! number of components
real*8 ,parameter :: rg = 8.313999999999999d0 ! ideal gas constant
c       real*8 ,parameter :: trf= 0.0d0        ! reference temperature
real*8 ,parameter :: trf= 273.15d0           ! reference temperature

```

### C.3.11 thermo\_data.dat

```

tolerance (Component order: cyclohexane, n-Heptane, toluene)
1.000d-7

Mw(kg/mol)  Tc(K)  Tboil(K)  Pc(Pa)  w
84.141d-3   553.68d0  353.15d0  4.100d6  0.212d0
100.204d-3  540.20d0  371.55d0  2.736d6  0.346d0
92.141d-3   591.80d0  383.95d0  4.113d6  0.262d0

del(binary interaction parameters) ( for NC = 3 , 3x3 ; for NC = 4 , 4x4 )
0.0d0      0.0d0  0.0d0
0.0d0  0.0d0  0.0d0  0.0d0
0.0d0  0.0d0  0.0d0
  cenh1   cenh2   cenh3   cenh4(J/molK)
-66.672040d0 68.843536d-2 -38.50535d-5 80.62568d-9
-12.717000d0 70.802000d-2 -40.14000d-5 90.00000d-9
-34.363192d0 55.885688d-2 -34.43432d-5 80.33280d-9

```

## APPENDIX D

### STUDIED CHEMICALS

**Table D.1** Specifications for Hydrocarbons.

|   | <b>Cyclohexane</b>         | <b>n- Heptane</b>          | <b>Toluene</b>             |
|---|----------------------------|----------------------------|----------------------------|
| Boiling Temp(@ 1 atm)                   | 353.85 K                   | 371.55 K                   | 383.75 K                   |
| Melting Temp                            | 279.75 K                   | 182.55 K                   | 178.15 K                   |
| MW (g /mole)                            | 84.161                     | 100.204                    | 92.141                     |
| Heat Capacity Coefficient, (J / mole.K) |                            |                            |                            |
| $C_p^a$                                 | -66.672040                 | -12.717000                 | -34.363192                 |
| $C_p^b$                                 | $68.843536 \times 10^{-2}$ | $70.802000 \times 10^{-2}$ | $55.885688 \times 10^{-2}$ |
| $C_p^c$                                 | $-38.50535 \times 10^{-5}$ | $-40.14000 \times 10^{-5}$ | $-34.43432 \times 10^{-5}$ |
| $C_p^d$                                 | $80.62568 \times 10^{-9}$  | $90.00000 \times 10^{-9}$  | $80.33280 \times 10^{-9}$  |
| $T_c$ , K                               | 553.58                     | 540.2                      | 591.8                      |
| $P_c$ (Pa)                              | $4.10 \times 10^{-6}$      | $2.72 \times 10^{-6}$      | $4.10 \times 10^{-6}$      |
| $V_c$ (m <sup>3</sup> / kmole)          | 0.308                      | 0.428                      | 0.314                      |
| $Z_c$                                   | 0.274                      | 0.259                      | 0.262                      |
| w                                       | 0.212                      | 0.346                      | 0.262                      |



**Table D.2** List of Correlations Used to Predict Physical Properties of the Hydrocarbons.

| Physical property that will be predicted                             | Author(s)                | Correlation(s)  | Equation number | Error  |
|--|--------------------------|---|-----------------|--|
| Gas viscosity of pure hydrocarbons at low pressure (below Tr of 0.6) | Stiel and Thodos (1961)  | $\mu = 4.60 \times 10^{-7} \frac{NMW^{1/2} P_c^{2/3}}{T_c^{1/6}}$   | (D.1)           | The average error is about 3 percent for hydrocarbons below ten carbon atoms. The percentage of the error increases to 5 – 10 percent for heavier hydrocarbons |
|  |                          | $N = 0.0003400 T_r^{0.94} \text{ for } T_r \leq 1.5$  | (D.2)           |  |
|  |                          | $N = 0.0001778(4.58 T_r - 1.67)^{0.625} \text{ for } T_r > 1.5$   | (D.3)           |  |
| Gas viscosity for multicomponent mixtures of hydrocarbons            | Bromley and Wilke (1951) | $\mu_m = \sum_{i=1}^{NC} \frac{\mu_i}{1 + \sum_{\substack{j=1 \\ j \neq i}}^{NC} \left( Q_{ij} \frac{x_j}{x_i} \right)}$  | (D.4)           | Error related to that correlation is about 3 percent.  |
|  |                          | $Q_{ij} = \frac{1 + \left[ \left( \frac{\mu_i}{\mu_j} \right)^{1/2} \left( \frac{\mu_j}{\mu_i} \right)^{1/4} \right]^2}{\sqrt{8} \left[ 1 + \frac{MW_i}{MW_j} \right]^{1/2}}$ | (D.5)           |  |

**Table D.2** List of Correlations Used to Predict Physical Properties of the Hydrocarbons (Cont'd).

| Physical property that will be predicted     | Author(s)                 | Correlation(s)  | Equation number | Error  |
|--|---------------------------|---|-----------------|--|
|  |                           | $\log(\mu * 0.001) = B \left( \frac{1}{T} - \frac{1}{T_o} \right)$                    | (D.6)           |  |
|  |                           | $N^o = N + \sum \Delta N_i$   | (D.7)           |  |
| Liquid viscosity of pure hydrocarbon liquids | van Velzen et al.(1972)   | $N^o \leq 20 \quad T_o = 28.86 + 37.439N^o - 1.3547N^{o2} + 0.02076N^{o3}$            | (D.8)           | Errors are average 15 percent for several compounds. |
|  |                           | $N^o > 20 \quad T_o = 8.164N^o + 238.59$  | (D.9)           |  |
|  |                           | $B = B_a + \sum \Delta B_i$   | (D.10)          |  |
|  |                           | $N^o \leq 20 \quad B_a = 24.79 + 66.885N^o - 1.3173N^{o2} - 0.00377N^{o3}$            | (D.11)          |  |
|  |                           | $N_o > 20 \quad B_a = 530.59 + 13.740N^o$   | (D.12)          |  |
| Liquid viscosity of hydrocarbon mixtures     | Kendall and Monroe (1917) | $\mu_m = \left( \sum_{i=1}^{NC} x_i \mu_i^{1/3} \right)^3$                            | (D.13)          | Errors change for 3 – 15 percent.                    |
| Surface tension of a component               | Brock and Bird (1955)     | $\sigma = 4.601 \times 10^{-4} P_c^{2/3} T_c^{1/3} Q(1 - T_r)^{11/9}$                 | (D.14)          | Errors in that correlation are less than 5 percent.  |
|  |                           | $Q = 0.1207 \left( 1 + \frac{T_{br} (\ln P_c - 11.5261)}{1 - T_{br}} \right) - 0.281$ | (D.15)          |  |

**Table D.2** List of Correlations Used to Predict Physical Properties of the Hydrocarbons (Cont'd).

| Physical property that will be predicted  | Author(s)                            | Correlation(s)  | Equation number | Error  |
|---|--------------------------------------|---|-----------------|--|
| Surface tension for mixture   | Winterfeld, Scriven and Davis (1978) | $\sigma_m = \left( \sum_{i=1}^n \sum_{j=1}^n P^2 \left( \frac{x_i}{\rho_{Li}} \right) \left( \frac{x_j}{\rho_{Lj}} \right) (\sigma_i \sigma_j)^{(1/2)} \right)$ | (D.16)          | Error is about 3 - 5 percent.                                |
|   |                                      | $\frac{1}{P} = \sum_{i=1}^n \frac{x_i}{\rho_{Li}}$  | (D.17)          |  |
| Gas diffusivity of binary hydrocarbon – hydrocarbon gas systems at low pressures (below 500 psia = 3.5 MPa) | Gilliland (1934)                     | $D_{12} = \frac{0.1014T^{1.5} \left( \frac{1}{MW_1} + \frac{1}{MW_2} \right)^{0.5}}{\rho(V_1^{1/3} + V_2^{1/3})^2}$   | (D.18)          | The error of the method of Gilliland is less than 4 percent. |
| Molar volume at the normal boiling point  | Tyn and Calus (1975)                 | $V_i = 0.285V_{ci}^{1.048}$   | (D.19)          | Not stated   |
| Gas phase diffusion coefficients in multicomponent hydrocarbon systems                                      | Wilke (1950)                         | $D_m = \frac{1 - y_i}{\sum_{j \neq i} y_j / D_{ij}}$  | (D.20)          | Errors are about 5 percent                                   |

**Table D.3** Physical Properties of Ethanol and Water Utilized in the Simulation Runs.

| Physical Property                        | EtOH                    | Water                | Temperature [K] | Reference              |
|--|-------------------------|----------------------|-----------------|------------------------|
| <b>Gaseous Viscosity</b><br>[kg / (m.s)] | $8.6088 \times 10^{-3}$ | $125 \times 10^{-4}$ | 293.15          | Perry et. al.,<br>1997 |
| <b>Liquid Viscosity</b><br>[kg / (m.s)]  | $1.7840 \times 10^{-3}$ | $1 \times 10^{-3}$   | 293.15          | Perry et. al.,<br>1997 |
| <b>Surface Tension</b><br>[mN / m]       | 22.39                   | 72.75                | 293.15          | Kaye, 1969             |

**Table D.4** Specifications for Ethanol and Water.

|   | Ethanol                  | Water                   |
|---|--------------------------|-------------------------|
| Boiling Temp(@ 1 atm)                   | 351.6 K                  | 373.15 K                |
| Melting Temp                            | 158.8 K                  | 273.15 K                |
| MW (g /mole)                            | 46.069                   | 18.015                  |
| Heat Capacity Coefficient, (J / mole.K) |                          |                         |
| $C_p^a$                                 | 19.875                   | 32.218                  |
| $C_p^b$                                 | $20.946 \times 10^{-2}$  | $0.192 \times 10^{-2}$  |
| $C_p^c$                                 | $-10.372 \times 10^{-5}$ | $1.055 \times 10^{-5}$  |
| $C_p^d$                                 | $20.042 \times 10^{-9}$  | $-3.593 \times 10^{-9}$ |
| $T_c$ , K                               | 513.92                   | 647.13                  |
| $P_c$ (Pa)                              | $6.12 \times 10^{-6}$    | $21.94 \times 10^{-6}$  |
| $V_c$ (m <sup>3</sup> / kmole)          | 0.168                    | 0.056                   |
| $Z_c$                                   | 0.240                    | 0.228                   |
| w                                       | 0.643                    | 0.343                   |

**Table D.5** NRTL Model Parameters.

| <b>Component i</b>          | <b>EtOH</b>           | <b>H<sub>2</sub>O</b> |
|-----------------------------|-----------------------|-----------------------|
| <b>Component j</b>          | <b>H<sub>2</sub>O</b> | <b>EtOH</b>           |
| $\Delta g_{ij}$ (cal / mol) | -175.0164             | 1440.3479             |
| $\alpha_{ij}$               | 0.2959                | 0.2959                |

**Table D.6** Constants for Antoine Equation.

|                | <b>A<sub>1</sub></b> | <b>A<sub>2</sub></b> | <b>A<sub>3</sub></b> |
|----------------|----------------------|----------------------|----------------------|
| <b>Ethanol</b> | 5.33675              | 1648.220             | 230.918              |
| <b>Water</b>   | 5.11564              | 1687.537             | 230.170              |

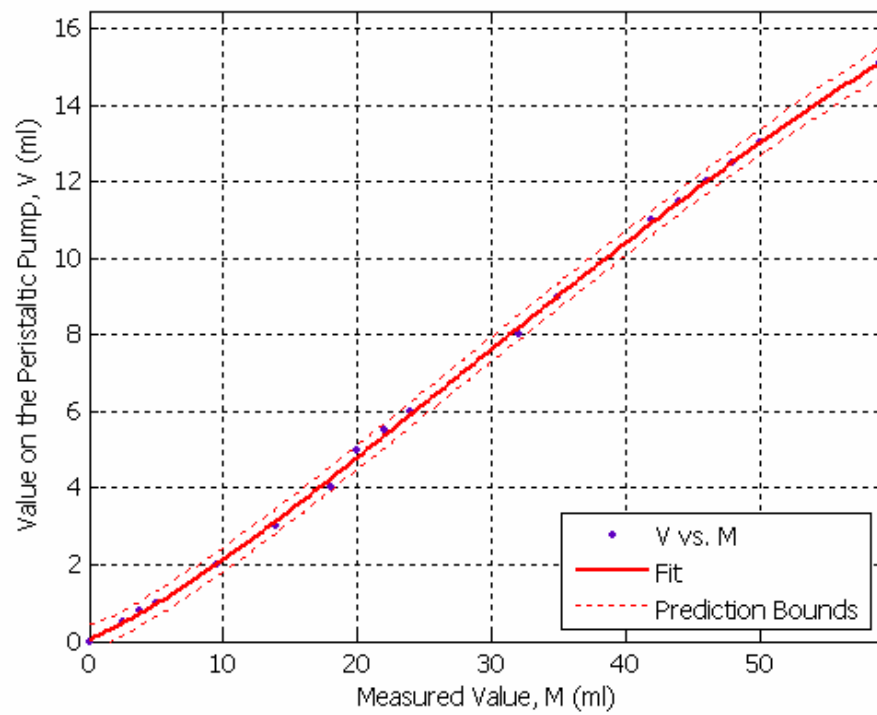
**Table D.7** Constants for Wagner Equation.

|                | <b>W<sub>1</sub></b> | <b>W<sub>2</sub></b> | <b>W<sub>3</sub></b> | <b>W<sub>4</sub></b> |
|----------------|----------------------|----------------------|----------------------|----------------------|
| <b>Ethanol</b> | -8.51838             | 0.34163              | -5.73683             | 8.32581              |
| <b>Water</b>   | -7.76451             | 1.45838              | -2.77580             | -1.23303             |

## APPENDIX E

### DETAILS OF EXPERIMENTAL SET UP

#### E. 1. Calibration Curve of Peristaltic Pumps



**Figure E.1** Calibration Curve of the Peristaltic Pumps.

Value on the peristaltic pump is estimated by using the calibration equation of the peristaltic pump which is given as follows,

$$V = -7 * 10^{-9} M^5 + 10^{-6} M^4 - 0.0001 M^3 + 0.0055 M^2 + 0.1693 M \quad (E.1)$$

## E. 2. Details for the Analysis of Compound Compositions

Samples are analyzed by using Poropak - Q column at gas chromatography (GC) (Hewlett – Packard 5890 Series II). Used temperatures for GC and areas obtained for samples, whose compositions are known, are tabulated at Table E.1 and E.2, respectively.

**Table E.1** Specifications for GC.

| Temperature (°C) |     |
|------------------|-----|
| Oven             | 150 |
| Injection        | 170 |
| Det A            | 180 |

**Table E.2** Areas Obtained for Prepared Samples.

| Sample Percentages              | Area  |         |
|---------------------------------|-------|---------|
|                                 | Water | Ethanol |
| 15% EtOH – 85% H <sub>2</sub> O | 26542 | 7624    |
| 30% EtOH – 70% H <sub>2</sub> O | 19890 | 11586   |
| 50% EtOH – 50% H <sub>2</sub> O | 12736 | 13865   |
| 60% EtOH – 40% H <sub>2</sub> O | 10050 | 14927   |
| 70% EtOH – 30% H <sub>2</sub> O | 7664  | 15900   |

Compositions of the samples taken during experiment are calculated by using Equations E.2 and E.3.

$$X_{EtOH} = \frac{\kappa_{EtOH} \beta_{EtOH}}{\kappa_{EtOH} \beta_{EtOH} + \kappa_{H_2O} \beta_{H_2O}} \quad (E.2)$$

$$x_{H_2O} = \frac{\kappa_{H_2O} \beta_{H_2O}}{\kappa_{EtOH} \beta_{EtOH} + \kappa_{H_2O} \beta_{H_2O}} \quad (E.3)$$

Since ethanol is more volatile than water, ethanol is chosen as a base component. Therefore, base component factor of the ethanol is equal to 1 as follows,

$$\beta_{EtOH} = 1 \quad (E.4)$$

To calculate base component factor of water,  $\beta_{H_2O}$  Equation E.2 and E.3 are analyzed with information tabulated at Table E.2. For each sample with known compositions, only unknown value is base component factor of water in Equations E.2 and E.3.

**For sample 15%EtOH and 85%H<sub>2</sub>O**

$$0.15 = \frac{(7624)(1)}{(7624)(1) + (26542)\beta_{H_2O}} \Rightarrow \beta_{H_2O} = 1.6277$$

**For sample 30%EtOH and 70%H<sub>2</sub>O**

$$0.30 = \frac{(11586)(1)}{(11586)(1) + (19890)\beta_{H_2O}} \Rightarrow \beta_{H_2O} = 1.35917$$

**For sample 50%EtOH and 50%H<sub>2</sub>O**

$$0.50 = \frac{(13865)(1)}{(13865)(1) + (12736)\beta_{H_2O}} \Rightarrow \beta_{H_2O} = 1.088$$

**For sample 60%EtOH and 40%H<sub>2</sub>O**

$$0.60 = \frac{(14927)(1)}{(14927)(1) + (10050)\beta_{H_2O}} \Rightarrow \beta_{H_2O} = 0.9901$$

**For sample 70%EtOH and 30%H<sub>2</sub>O**

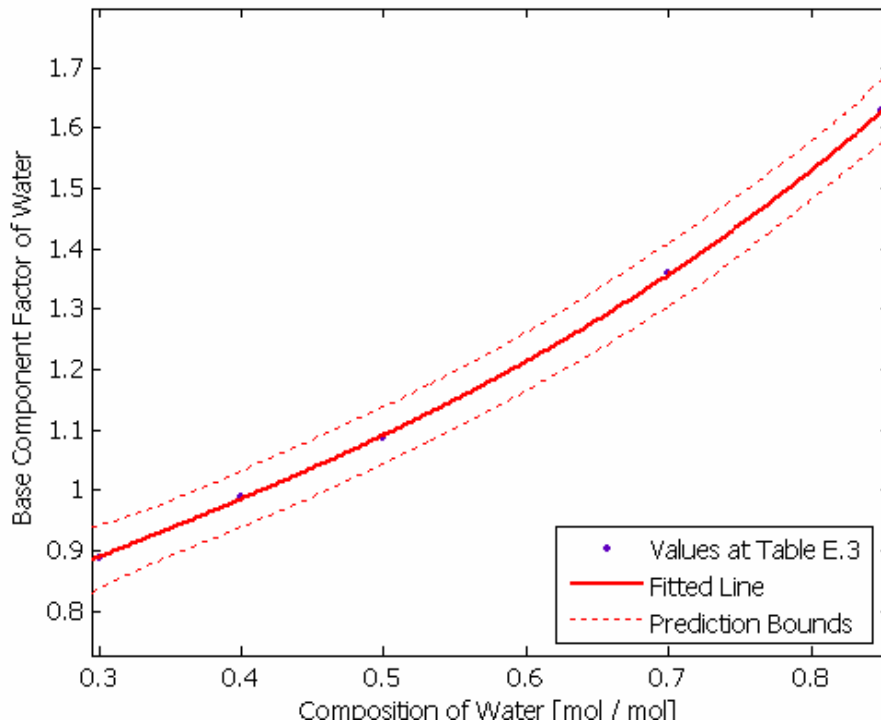
$$0.70 = \frac{(15900)(1)}{(15900)(1) + (7664)\beta_{H_2O}} \Rightarrow \beta_{H_2O} = 0.8891$$

$\beta_{H_2O}$  shows parabolic behavior with respect to water composition. The data is given at Table E.3 and behavior can be observed by Figure E.2.



**Table E.3** Data for Base Component Factor of Water.

| $x_{H_2O}$ | $\beta_{H_2O}$ |
|------------|----------------|
| 0,85       | 1,6277         |
| 0,7        | 1,35917        |
| 0,5        | 1,088          |
| 0,4        | 0,9901         |
| 0,3        | 0,8891         |



**Figure E.2** Behavior of Base Component of Water.

By fitting the values at Table E.3, following equation is obtained to estimate base component of water,

$$\beta_{H_2O} = 1.0152x_{H_2O}^2 + 0.1611x_{H_2O} + 0.7545 \quad (E.5)$$

Then, Equation E.2 can be rewritten as follows:

$$x_{\text{EtOH}} = \frac{\kappa_{\text{EtOH}} \beta_{\text{EtOH}}}{\kappa_{\text{EtOH}} \beta_{\text{EtOH}} + \kappa_{\text{H}_2\text{O}} (1.0152 x_{\text{H}_2\text{O}}^2 + 0.1611 x_{\text{H}_2\text{O}} + 0.7545)} \quad (\text{E.6})$$

and,

$$x_{\text{EtOH}} + x_{\text{H}_2\text{O}} = 1 \quad (\text{E.7})$$

By rearranging Equation E.6, following equation is obtained to estimate liquid composition of ethanol.

$$1,0152 \kappa_{\text{H}_2\text{O}} x_{\text{EtOH}}^3 - 2,1915 \kappa_{\text{H}_2\text{O}}^2 + (\kappa_{\text{EtOH}} + 1,9308 \kappa_{\text{H}_2\text{O}}) x_{\text{EtOH}} - \kappa_{\text{EtOH}} = 0 \quad (\text{E.8})$$