Improvement of PGB Depropanizer Column Model by Removing Ideal Mixture Assumption

by

Mohd Hirmi Bin Mat Sidek

A project dissertation submitted to the
Chemical Engineering Programme
Universiti Teknologi Petronas
In partial fulfillment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(CHEMICAL ENGINEERING)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
NOVEMBER, 2004

CERTIFICATION OF APPROVAL

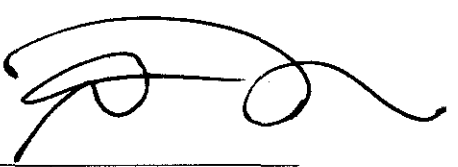
Improvement of PGB Depropanizer Column Model by Removing Ideal Mixture Assumption

by

Mohd Hirmi Bin Mat Sidek

A project dissertation submitted to the
Chemical Engineering Programme
Universiti Teknologi Petronas
In partial fulfillment of the requirement for the
BACHELOR OF ENGINEERING (Hons)
(CHEMICAL ENGINEERING)

Approved by,

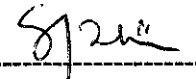


(Mr. Nooryusmiza Yusoff)
Main Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
NOVEMBER, 2004

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken by unspecified sources or persons.



(MOHD HIRMI B MAT SIDEK)

ABSTRACT

This report entitled "Improvement of PGB Depropanizer Column Model by Removing Ideal Mixture Assumption" is basically a dissertation report for the Chemical Engineering Final Year Research Project. Since a long time ago, chemical engineers have been dealing with a lot of optimization methods for distillation column. The optimization is crucial because the best parameters are needed so that the desired product of distillation column can be achieved and also saving energy consumed in the plant which automatically saves million of dollars for the plant. The main purpose of this research is to improve the de-propanizer column modeling by removing the ideal mixture assumption made in the early stage of this project. The results obtained are compared with the results obtained from HYSYS by using the plant data of PGB De-Propanizer Column. The final objective of this project is to show the results in the form of graphical user interface, GUI. The modeling for the non-ideal mixture was done by using the MATLAB 7.0 software. First of all, mass balance was done by using the narrow feed boiling procedure and it was converged by applying the theta-convergence method. Then, the energy balance calculation was taken into consideration by calculating the energy balance for ideal system and non-ideal system. Singular Value Decomposition (SVD) method was used for the convergence of matrix. Comparison of results between results obtained from MODEL with the results obtained from HYSYS was made. The results were presented in the form of graphical user interface GUI for more users friendly. Several results were obtained for mass balance and energy balance and discussed in the discussion section. Some recommendations were made for future improvement of the model. This project is crucial as optimization of distillation column is one of the most important features to be achieved by the engineers in any plant. This is because any changes in the data and operating parameters such as the temperature and pressure of the column will results to the difference in compositions of product and by-product. It is hoped that this model will be used as a tool for engineers to obtain desired parameters and would predict the results of the actual distillation column by referring to the model.

ACKNOWLEDGEMENT

The author of this report would like to thank all the parties that are involved in accomplishing this research study, directly or indirectly. First of all thank you to Allah, God Almighty for making this project possible, to the author's parents who have always being supportive during the project undergoing. The highest gratitude goes to the author's supervisor, Mr Nooryusmiza Yusoff, Lecturer, Chemical Engineering Department of University Teknologi Petronas, for all the guidance and help throughout this research. Million of thanks to Brother Khirul Akmal b Azahri, Undergraduating student of UTP, who have been very co-operative especially in helping and sharing his expertise and knowledge with the author in conducting the modeling using the MATLAB and also the GUI. Not forgotten to the author's entire colleague for being supportive and thoughtful in sharing their ideas. This acknowledgement also dedicated to the other parties that contribute towards the accomplishment of this research study.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND NOMENCLATURES

ABBREVIATIONS

UTP	Universiti Teknologi PETRONAS
PGB	PETRONAS Gas Berhad
GUI	Graphical User Interface
SVD	Singular Value Decomposition
SSE	Sum of square error
RMSE	Root mean square error

NOMENCLATURES

T	Temperature
P	Pressure
Z	Compressibility factor
V	Volume
K	Equilibrium Constant

CHAPTER 1

INTRODUCTION

1.0 INTRODUCTION

1.1 Background of Study

Distillation is a separation process by which mixtures can be divided into two or more fractions that have higher purity than the original mixture. It has been used since quite a long time ago and it still is the most important separation process. It is usable when the volatility of the components differs sufficiently.

Chemical Engineers generally started research on distillation already in 60'ies when experimental efficiencies of valve and bubble cap trays were published. Pilot scale column was erected at the end of that decade. The column was equipped with on-line analyzer system based on mass spectrometer. That system enabled transient vapor and liquid sampling. The experimental results from the system have been used to develop and verify transient and steady state mathematical models and design methods for distillation. Rigorous multicomponent distillation program was developed then at 70'ies

Technique of separation of mixture has been developed hundred years ago by the engineers. One of the most popular techniques used is by using distillation column. Distillation has been around for a long time. Earliest references are to Maria the Jewess who invented many types of stills and reflux condensers. Common Middle Ages and Renaissance uses of distillation included the manufacture of brandy and other spirits from wine. Another early use was the manufacture of perfumes and essences. Other early users of distillation include the Alchemists. Of course, the history of distillation does not end there. Today we use it for more than just spirits. Nowadays, distillation column has been one of the most important equipment in the plant. Many industries use distillation for critical separations in making useful

products. These industries include petroleum refining, beverages, chemical processing, petrochemicals, and natural gas processing. Natural gas processing started using distillation in the early 1900's. From time to time, research and tests were made in order to improve the efficiency of the distillation. Several methods and calculations were developed by engineers to achieve the best separation of mixture using distillation column. Distillation can be considered as complex equipment. Known as a major energy consumer, distillation column is also highly dependent on the operating conditions, the feed composition and type of columns used. Any significance change on any of it might results to difference in column efficiency thus reflects the production of desired product. For that matter, the student has taken this challenge to do the project which tries to improve and optimize the distillation column at PGB Kerteh by removing the ideal mixture assumption.

1.2 Problem Statement

1.2.1 Problem Identification

The title given for this project is "Improvement of PGB Depropanizer Column Model by Removing Ideal Mixture Assumption". Basically this project aimed to model unit operation specifically for depropanizer column using mathematical engineering software MATLAB. Verification of the results will be done by comparing it with other simulation software available, namely HYSYS. Further improvement of programming coding will be done for the deethanizer and debutanizer column and verification of results will also be made by comparing with HYSYS. A GUI (Graphic User Interface) will be used as an output to display all data for the distillation column.

1.3 Significant of Project

1.3.1 To PGB

The project will provide a helpful tool for PGB for the optimization of depropanizer, deethanizer and debutanizer column. Since the previous student had done with the optimization of depropanizer column using the ideal mixture assumption, this project will gives PGB an opportunity to optimize their plant production as two methods of simultaneous multicomponent convergence can be compared and the best results can be chosen.

1.3.2 To the student

The student will gain many knowledge about the distillation column, especially with the non ideal mixture procedure, where this assumption has been used by the engineers in the plant for their distillation column optimization. This project gives an opportunity to familiarize with many engineering software such as the MATLAB programming tools, and HYSYS. As these software's are the most used software in the chemical engineering world, it will give the student the knowledge about these software to a greater extend, and the student will not feel awkward when entering the real working environment that utilize these software in every project they involved in.

1.4 Objective of study

- To model a depropanizer distillation column by removing the ideal mixture assumption using MATLAB software.
- Develop a modeling for de-ethanizer and de-butanizer column.
- To compare the results obtained from the model with the results from HYSYS.
- Show the output of distillation column by using the GUI (Graphic User Interface) and do some improvement to the GUI

1.5 Scope of Study

- Using mathematical equations of the distillation column especially for the calculation for non ideal mixtures.

- Develop modeling by using MATLAB based on the mathematical equations obtained.

- Comparison of results obtained from MATLAB with HYSYS.

1.6 Relevancy of Project

1.6.1 Knowledge gained from the project

The project is basically a continuation from the previous student's project, such that, the project will try to optimize the multicomponent distillation column of de-propanizer using the non ideal mixture calculations. For that, the student must obtain as many information as possible regarding the previous project to ensure that the student know what to optimize and how to do the project. Further readings and net search will also help the student to know much more about the project such as the distillation column calculations and many more. This will help the student to gain more knowledge in chemical engineering field especially when dealing with distillation column.

The results from the project will be presented in Graphical User Interface (GUI) which had never been learnt by the student before. This is a very good opportunity for the student to explore and understand the bond between chemical engineering field and multimedia, and appreciate it.

1.6.2 Close relationship between university and industry

The project is mainly trying to optimize the production of light key in distillation column, be it de-propanizer, de-butanizer and de-ethanizer at PGB. As for that matter, the student is required to gain as many data as possible from PGB so that a comparable data between the data obtained from modeling and actual plant data can be made. This will help to close the relationship between University Teknologi Petronas and PGB as they work together in optimizing the distillation column, resulting to a better production of desired product.

1.6.3 Economic point of view

Optimizing the distillation column may lead to more product recovery at the top and heavier key component in the bottom. As example, in a de-propanizer column, more propane recovered in the top will results to more butane obtained in the bottom which is more purified to be sent to the de-butanizer column. This might results to better separation and thus become an important key aspect in production of desired product. There comes the economic point of view where higher productions of products will results to higher revenue and cost savings.

1.7 Feasibility of the Project

During the end of the semester, the student needs to come out with the results for the project. The main objective to be achieved is to complete the modeling for the calculation for the energy balance of non ideal mixtures. The student has been given approximately 14 Weeks to complete this project. With the help from the supervisor, Mr Yusmiza and complete guidance from lecturers and reference, the project can be completed within the timeframe.

CHAPTER 2

LITERATURE REVIEW

2.0 LITERATURE REVIEW

2.1 Distillation column

2.1.1 General Principles

Separation operations achieve their objective by the creation of two or more coexisting zones which differ in temperature, pressure, composition, and phase state. Each molecular species in the mixture to be separated reacts in a unique way to differing environments offered by these zones. Consequently, as the system moves toward equilibrium, each species establishes a different concentration in each zone, and this result in a separation between species. The separation called distillation utilizes vapor and liquid phases at essentially the same temperature and pressure for the coexisting zones. Various kinds of devices such of random or structured packings and plats or trays are used to bring the two phases into intimate contact.

The feed material, which is to be separated into fractions, is introduced at one or more points along the column shell. Because of the difference in gravity between vapor and liquid phases, liquid runs down the column, cascading from tray to tray, while vapor flows up the column, contacting the liquids at each tray. Liquid reaching the bottom of the columns is partially vaporized in a heated reboiler to provide boil-up, which is sent back up the column. The remainder of the bottom liquid is withdrawn as bottoms, or bottom product. Vapor reaching the top of the column is cooled and condensed to liquid in overhead condenser. Part of this liquid is returned in the column as reflux to provide liquid overflow. The remainder of the overhead stream is withdrawn as distillate, or overhead product. In some cases only part of the vapor is condensed so that a vapor distillate can be withdrawn.

This overall pattern in a distillation column provides counter-current contacting vapor and liquid streams on all the trays through the column. Vapor and liquid phases on a given tray approach thermal, pressure, and composition equilibriums to an extent dependent upon the efficiency of the contacting tray. The lighter components tend to concentrate in the vapor phase, while the heavier components tend toward the liquid phase. The result is a vapor phase that becomes richer in light components as it passes up the column and a liquid phase that becomes richer in heavy components as it cascades downward. The overall specification achieved between the distillate and bottoms depends primarily on the relative volatilities of the components, the number of contacting trays, and the ratio of liquid-phase flow rate to the vapor-phase flow rate.

If the feed is introduced at one point along the column shell, the column is divided in to an upper section, which is often referred to as the stripping section. These terms become rather indefinite in multiple feed columns and in columns which a liquid and vapor side stream is withdrawn somewhere along the column length in addition to the two end-product streams.

2.1.2 Distillation Main Components

Distillation column is made up of several components, each of which is used either to transfer heat energy or enhance material transfer. A typical distillation contains several major components:

- a vertical shell where the separation of liquid components is carried out
- column internals such as trays/plates and/or packings which are used to enhance component separations
- a reboiler to provide the necessary vaporisation for the distillation process
- a condenser to cool and condense the vapor leaving the top of the column

- a reflux drum to hold the condensed vapour from the top of the column so that liquid (reflux) can be recycled back to the column

The vertical shell houses the column internals and together with the condenser and reboiler, constitute a distillation column. A schematic of a typical distillation unit with a single feed and two product streams is shown below:

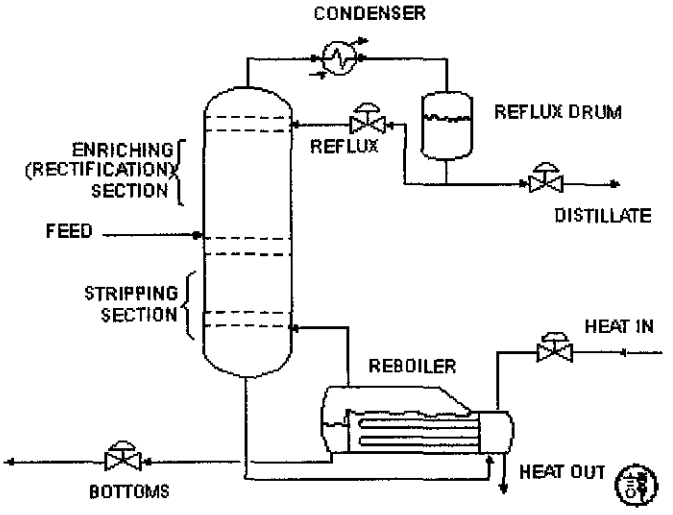


Figure 2:1: Distillation Column

2.2 Ideal mixtures vs. Non-ideal mixtures

The most restrictive thermodynamic model of a mixture is called the ideal mixture model, IMM that assumes that volumetric and energetic properties of a mixture are just the linear combination of those of their pure constituents (weighted with their relative proportions). In general, ideal mixture is a mixture where it is assumed that the individual molecules take up no volume and that they do not interact with each other.

All the components of an ideal mixture at a given T and P must be in the same phase when pure. For example at 15 °C and 100 kPa the nitrogen and oxygen in air is in the gas phase, and water and methanol in liquid phase.

On the other hand, non ideal mixtures deviate more or less from this simple model. Non-ideality is due to the interaction between molecules due to intermolecular forces. Many solutions are far from ideal because the interaction energy between molecules is not negligible as it would be for an ideal solution. Moreover these energies are complex functions of molecular properties and relative molecular orientations. In real multi-phase, multi-component processes it is often the case that we have multiple components in both the liquid and gas phases, such that the interaction energy between molecules must be taken into consideration. So, the best solution to be used when dealing with the multi-component processes is by using the laws of non ideal mixtures.

2.3 Fugacity

2.3.1 Methods for calculating fugacity

Several methods have been developed for calculating fugacity from measurement of pressure and molar volumes of non ideal mixture. It can be divided into two major groups, namely Graphical or Numerical method and Analytical method.

Graphical or Numerical method

- Using the α function
- Using the compressibility factor

Analytical methods

- Based on Virial Equation
- Based on the van der Waals equation of state
- An approximation method
- Peng Robinson equation of state

In this project, the Peng Robinson equation of state will be used to determine the value of fugacity of each component.

2.3.2 Fugacity for gaseous mixtures

Several methods are commonly used for estimating the fugacity of a species in a gaseous mixture. The most appropriate method is based on the observation that some gaseous mixtures follow Amagat's law, that is

$$V(T,P,N_1,N_2.....) = \sum N_i V_i(T,P) \text{ ----------(EQ 1)}$$

which, on multiplying by P/NRT can be written as

$$Z(T,P,y) = \sum y_i Z_i(T,P) \text{ ----------(EQ2)}$$

Here, Z = PV/nRT is the compressibility of the mixture. At higher pressure, the accurate equations of state must be used. One can, for example the Peng-Robinson equation of state

$$P = \frac{RT}{V-b} - \frac{a(T)}{V(V+b)+b(V-b)} \text{ ---------- (EQ3)}$$

where the parameters a and b are now for the mixtures. To obtain these mixture parameters one starts with the a and b parameters of pure components obtained from either fitting pure component data or the generalized correlation and then uses the mixing rules

$$a = \sum\sum y_i y_j a_{ij} \text{ ---------- (EQ4)}$$
$$b = \sum y_i b_i \text{ ---------- (EQ5)}$$

where aii and bi are the parameters for pure component i, and combining rule

$$a_{ij} = \sqrt{a_{ii} a_{jj}}(1-k_{ij}) = a_{ji} \text{ ----------(EQ6)}$$

Here, the value of k_{ij} is known as the binary interaction parameters. This parameter is found by fitting the equation of state to mixture data. As a result of the mole fraction dependence of the equation of state parameters, the pressure is a function of mole, or, alternatively, the number of moles of each species present. Further evaluation of the Peng-Robinson equation yields

$$\ln\frac{f_i(v)}{y_i P} = \frac{B_i}{B}(Z_v - 1) - \ln(Z_v - B) - \frac{A}{2\sqrt{2}B}\left[\frac{2\Sigma y_j A_{ij}}{A} - \frac{B_i}{B}\right]\ln\left[\frac{Z_v + (1+\sqrt{2})B}{Z_v + (1-\sqrt{2})B}\right] \quad \text{(EQ7)}$$

where again, $A = aP/(RT)^2$, $B = bP/RT$ and Z_v is the vapor phase compressibility factor.

Z is the value of compressibility factor, and is determined from equation

$Z = PV / RT$ ---------- (EQ8)

Or by using the cubic equation

$Z^3 - (1\text{-}B)\,Z^2 + (A - 3B^2 - 2B)\,Z - (AB - B^2 - B^2 - B^3) = 0$ ------- (EQ9)

2.3.3 Fugacity for liquid mixtures

The estimation of species fugacity in liquid mixture is done by two different ways, depending on the data available and the components in the mixture. For liquid mixtures involving only hydrocarbons and dissolved gases, such as nitrogen, hydrogen sulfide, and carbon dioxide, simple equation of state may be used to describe liquid state behavior. The Peng-Robinson equation of state is used for the fugacity of liquid mixtures thus the fugacity of each species in the mixture is, following the same development as for the gaseous mixtures, given by

$$\ln\frac{f_i(v)}{x_i P}=\frac{B_i}{B}(Z_l-1)-\ln(Z_l-B)-\frac{A}{2\sqrt{2}B}\left[\frac{2\Sigma x_j A_{ij}}{A}-\frac{B_i}{B}\right]\ln\left[\frac{Z_l+(1+\sqrt{2})B}{Z_l+(1-\sqrt{2})B}\right] \quad \text{(EQ10)}$$

where Z_l is the liquid root for the compressibility factor. Consequently, the calculation scheme for species fugacity in liquid mixture can be described by equations of state is similar to that for gaseous mixtures except that the liquid phase, rather than the vapor phase, compressibility factor is used in the calculations.

2.4 Graphical User Interface

A user interface is the point of contact or method of interaction between a person and a computer or computer program. It is the method used by the computer and the user to communicate information. The computer displays text and graphics on the computer screen and may generate sounds with a speaker. The user communicates with the computer using input devices such as a keyboard, mouse, trackball, drawing pad, and microphone. The user interface defines the look and feel of the computer, operating system, or application. Often a computer or a program is chosen on the basis of pleasing design and functional efficiency of its user interface.

A graphical user interface, or known as GUI is a user interface incorporating graphical objects in some way such as windows, icons, buttons, menus and text. Selecting or activating these objects in some way usually causes an action or change to occur. The most common activation method is to use a mouse or other pointing device to control the movement of a pointer on the screen, and to press a mouse button to signal object selection or some other action.

On the other hand, GUIs can be used to create very effective tools and utilities in MATLAB or to build interactive demonstrations of project work. Other reasons of the usage of GUIs are:

- To simplify the MATLAB function that will be used over and over again by using the menus, buttons or text boxes as input methods

- Writing a function for others to use, as GUI will make it more user friendly and easier to understand

- To create an interactive demonstration of a process, technique, or analysis method.

One might want to use or refer to the MATLAB coding but it might take some times to understand the coding, and it will be much more worst if the person does not have enough experience in programming. As such, it is hoped that the GUI will act like 'bridge' that connects between the user and the programming codes, making them to interact to each other much better. The GUI can also plot the graph from the results calculated in the programming codes either in 2-dimension or 3-dimension. The GUI can also be added with other programming or software such as a web template for assistance to the user to use the model. All in all, the need of using GUI is crucial especially for any work tasks that utilizes lots of programming languages in MATLAB.

CHAPTER 3

METHODOLOGY

3.0 METHODOLOGY

3.1 Procedure Identification

Below are the several steps of methodology to be used by the student during the project.

Step 1: Overview about the project and subjected theories

Learning the theories is crucial as a first step to be taken before proceeding with the project. This is to ensure that the student will understand about the objective of the project and enable to come out with an appropriate recommendation regarding the results obtained later. After that, the student is required to read through the previous project report done by the previous student to allow the student have some idea on what is the problem definition, the aim of this project, and also methods to complete the project.

Step II: Data gathering and collecting information about the project

As the student had completed the first step, the next step is to collect all relevant information and data especially those operational data from PGB. Other important information to be collected are the methods used for the project, any assumptions made, recommendations made by the previous student that must taken into consideration, and also other additional tasks assigned by the supervisor to the student.

Step III: Learn about the modeling made by previous student

Before proceeding with the modeling using MATLAB, the student is required to understand the syntax and programming language made by previous student, including the GUI method. This is to ensure that the student know about the programming syntax so that the student can be able to continue and modify the programming language by changing it to different mathematical methods. As for that, the first thing to do here is basically learn about the MATLAB itself, learning the syntax and other features of MATLAB to be used in this project.

Step IV: Prepare flowchart of method used

Flowchart of the method to be used is prepared upon familiarization with the modeling method.

Step V: Writing the program code using MATLAB

The program code is written using the MATLAB software with the reference of other program code done by previous student. It is advised that the program code is written and be tested from stage to stage to ensure working model is achieved in the end of the project.

Step VI: Test the model

The model is tested by comparing the model output with actual plant data. As assigned, the student must compare the results of de-ethanizer and de-butanizer column at PGB, Kerteh. The results from the model will also be compared with the results from HYSYS.

Step VII: Discussion and recommendation

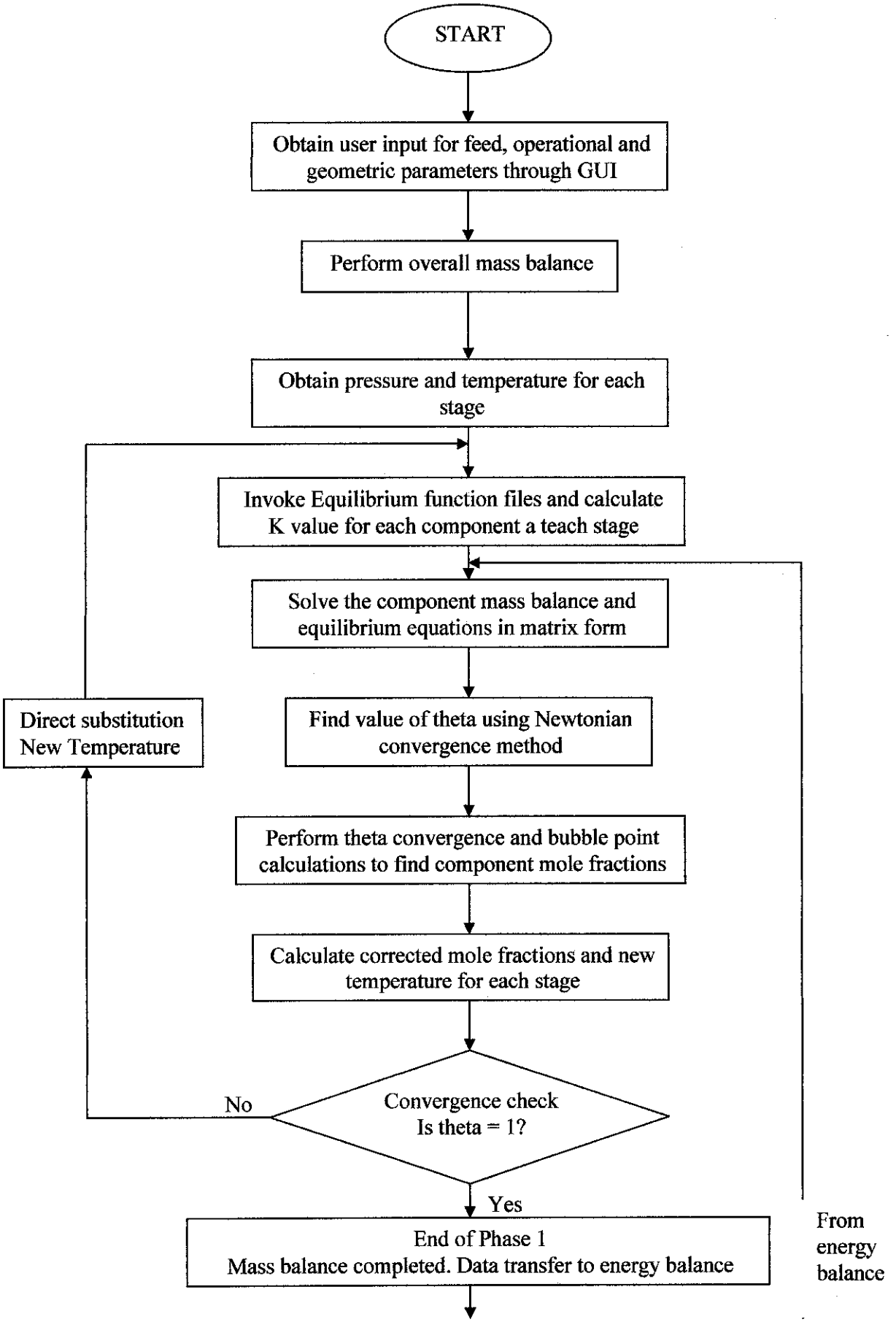
Discussions are made regarding the results obtained and any recommendations are also being suggested to improve the model.

3.2 Tools required

Software's that will be used during the project are:

1. MATLAB version 7.0
2. HYSYS Software
3. Microsoft Office (Microsoft Words, Excel)
4. Macromedia Flash MX

All these software are available in the UTP lab and workstation.


START
Obtain user input for feed, operational and geometric parameters through GUI
Perform overall mass balance
Obtain pressure and temperature for each stage
Invoke Equilibrium function files and calculate K value for each component a teach stage
Solve the component mass balance and equilibrium equations in matrix form
Find value of theta using Newtonian convergence method
Perform theta convergence and bubble point calculations to find component mole fractions
Calculate corrected mole fractions and new temperature for each stage
Convergence check
Is theta = 1?
No
Direct substitution
New Temperature
Yes
End of Phase 1
Mass balance completed. Data transfer to energy balance
From energy balance

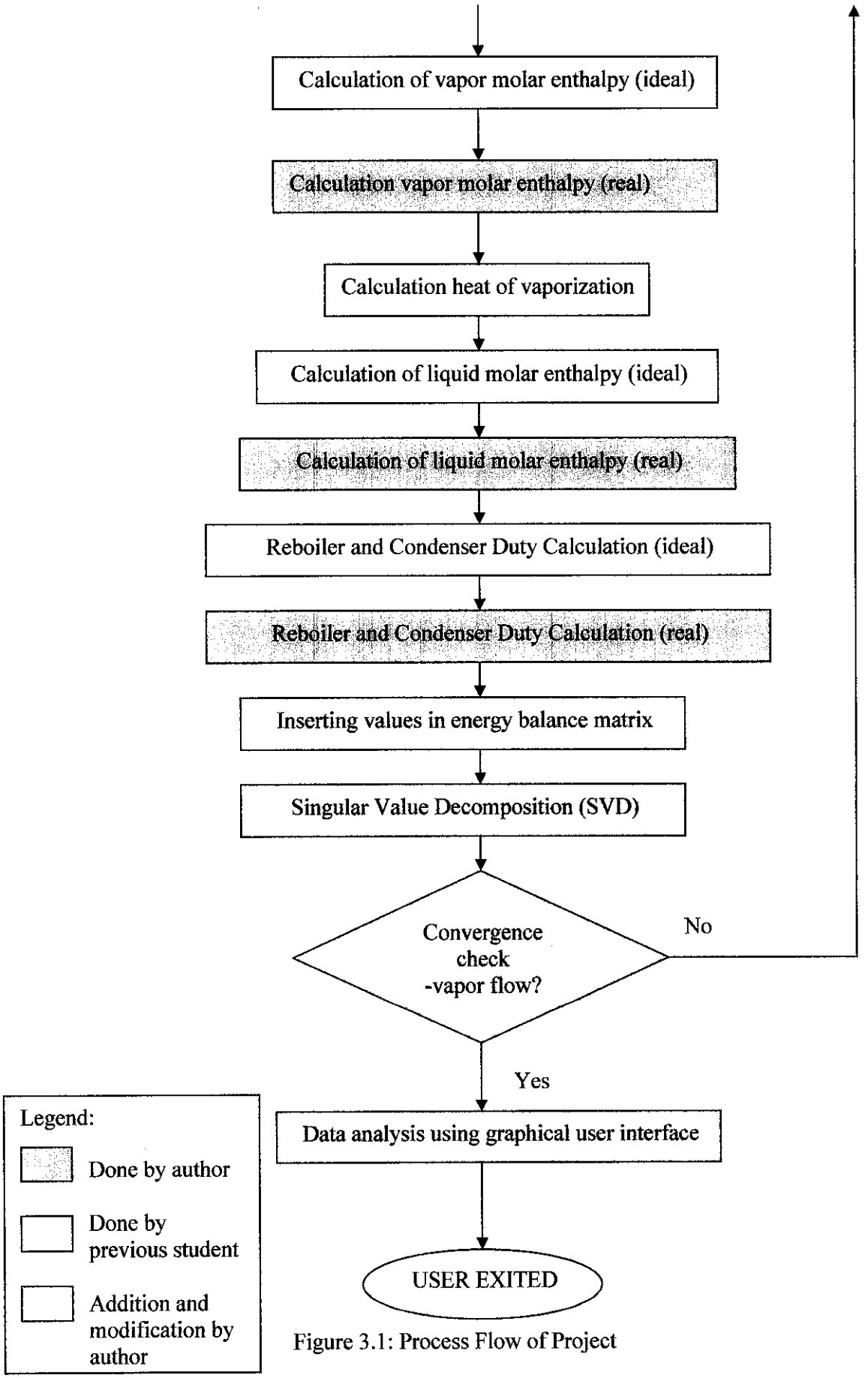


Figure 3.1: Process Flow of Project

CHAPTER 4

RESULTS AND DISCUSSION

4.0 RESULTS AND DISCUSSION

The results obtained are based on a constant set of input variables for the Non-Constant Molal Overflow model. The results and discussion will focus on the trends and profiles obtained for the separation parameters such as component mole fraction, enthalpy values and also reboiler and condenser duty. Generally, three kinds of graphs which is the graph obtained from Model for Ideal Mixtures, Model for Non Ideal Mixtures and from HYSYS will be shown for each result.

4.1 Input variables

The input parameters are shown in Table 4.1. The feed parameters are obtained from PGB, Kerteh. The operational data was obtained from plant data over the period of March 21st to March 25th 2004. The data available were at intervals of 2 hours duration. In this project, a single set of data was used in building the model and this data is the average of plant data available.

Table 4.1: Input variables from plant data

	Parameter	Value
Feed	Feed Rate	586 tonne/day
	Feed Composition	
	Propane	0.2094
	Isobutane	0.1559
	n-butane	0.2327
	Isopentane	0.1362
	n-pentane	0.0881
	n-hexane	0.1777
	Pressure	2050kPa
	Temperature	99.8°C
Operational	Reflux Ratio	5
	Distillate Flow Rate	93 tonne/day
	Recovery of Propane in D	0.98
	Condenser Temperature	45
	Condenser Pressure	1426kPa
	Reboiler Temperature	115°C
Geometric	Number of stages	53
	Feed stage	17

4.2 Input variables for HYSYS

Property package: Peng-Robinson

EOS Enthalpy method specification: Equation of state

Stage numbering: Bottom up

Table 4.2 : Input variables for HYSYS

Reflux Ratio	5	
Distillate Flow Rate	93 tonne/day	
Recovery of Propane in Distillate	0.98	
Condenser Temperature	45	
Condenser Pressure	1426kPa	
Reboiler Pressure	2100kPa	
Reboiler Temperature	115°C	
Geometric	Number of stages	53
	Feed stage	17

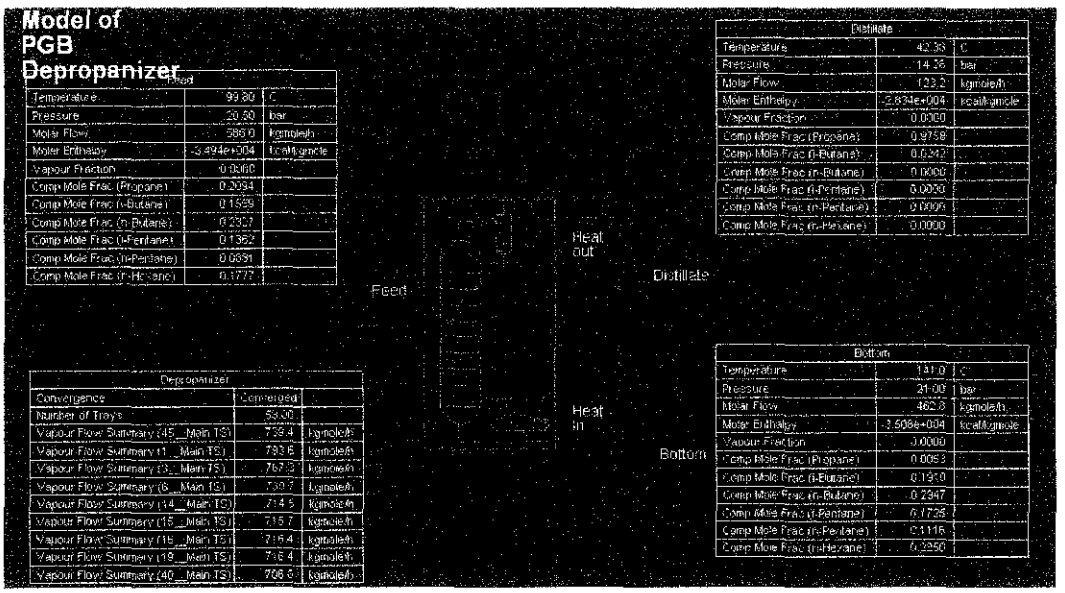


Figure 4.1: Converged HYSYS Simulation

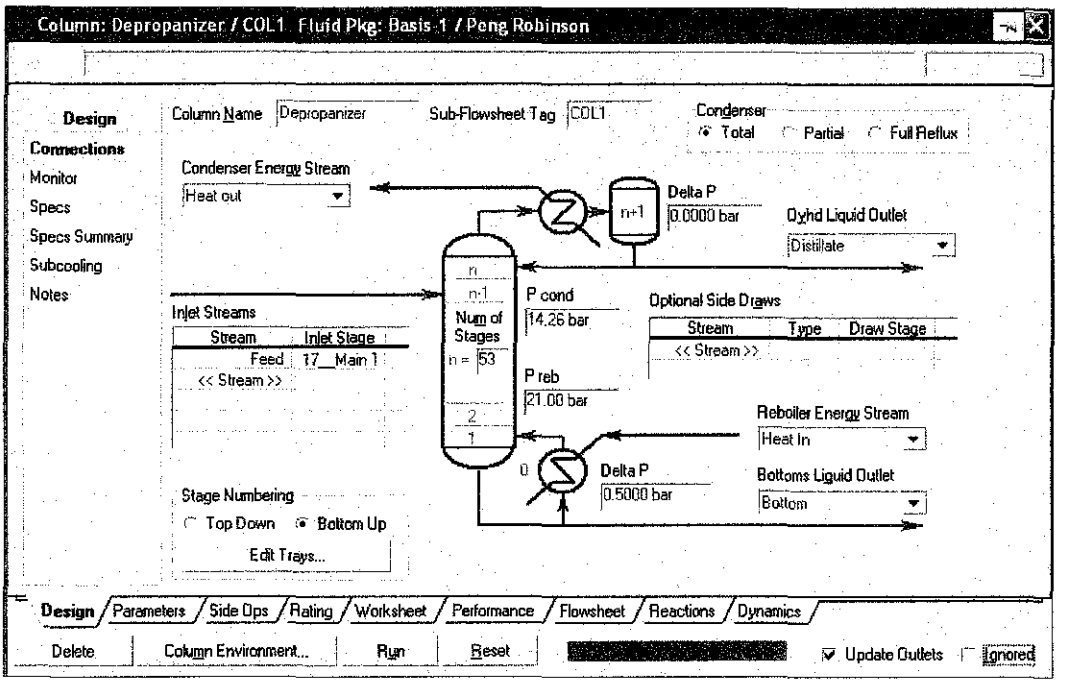


Figure 4.2: Design specifications for distillation column using HYSYS

4.3 Result of Mass Balance

4.3.1 Compositional Profiles

The objective in distillation is to obtain the product purity at the top outlet. In this model, the top product (propane) composition contains pure propane. This is due to the simplicity of the model. And it is the same with the other 28 stages below it. The compositional profile of propane with other components along the column given by the MODEL is shown in Figure 4.3.1(a). The compositional profile for propane given by HYSYS is shown in Figure 4.3.1(b). From the figures shown, the trend of compositional of propane along the column can be described as two increasing exponential curves in series that are connected at the feed stage. This is happen due to improper placing of the feed stage. Further explanation will be discussed with the discussion of temperature profile.

From this figure, it can be seen that the composition of propane increases as it proceed to higher column whereas the composition of remaining components decreases as it moved downwards to the bottom of the column. The compositional profile for isobutane and n-butane have a slight increase a bit as it moves down the column and then slowly reduces as it proceed to the lower stage column. This can be clearly explained from the characteristics of isobutane and n-butane as middle product which affected by the dominance of lighter key component which is propane at the top and also the dominance of heavy non key component which is hexane in the bottom. Thus, because of this characteristic of butane as the middle product, the trend of graph obtained is only slightly and coincidental.

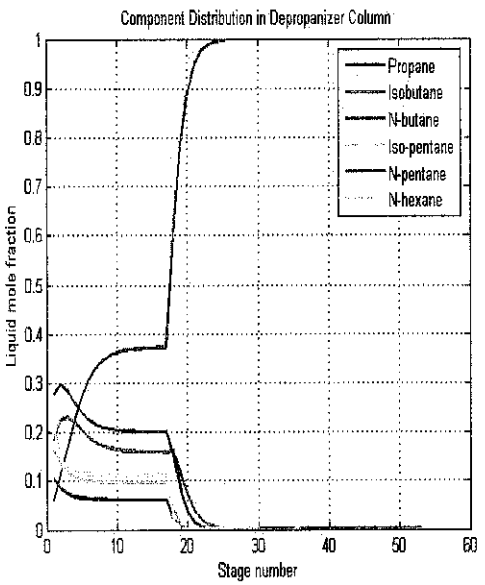


Figure 4.3.1(a) : Component Distribution on Depropanizer Column (MODEL)

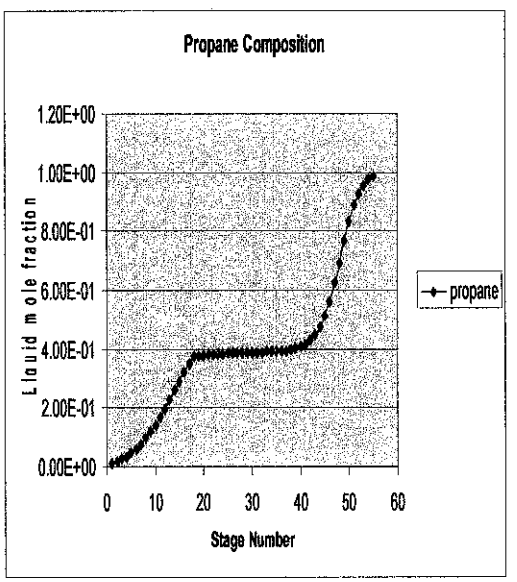


Figure 4.3.1(b) : Propane Distribution on Depropanizer Column (HYSYS)

4.3.2 K-values

The equilibrium constant, K is closely related to the temperature profile and also pressure of the column itself. The graph of distribution coefficient for each component is obtained from MODEL and HYSYS. The graphs are shown as below:

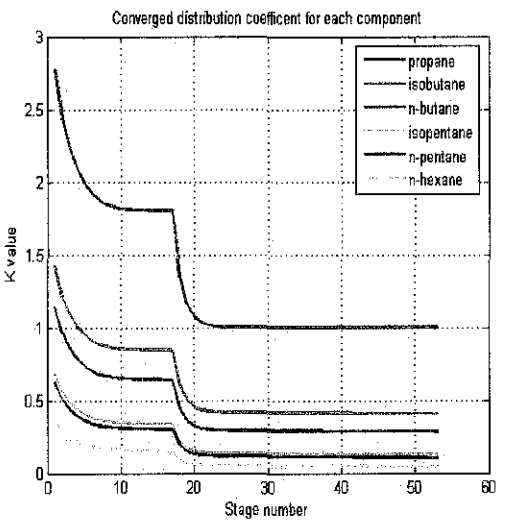


Figure 4.3.2(a) : Converged distribution coefficient for each component (MODEL)

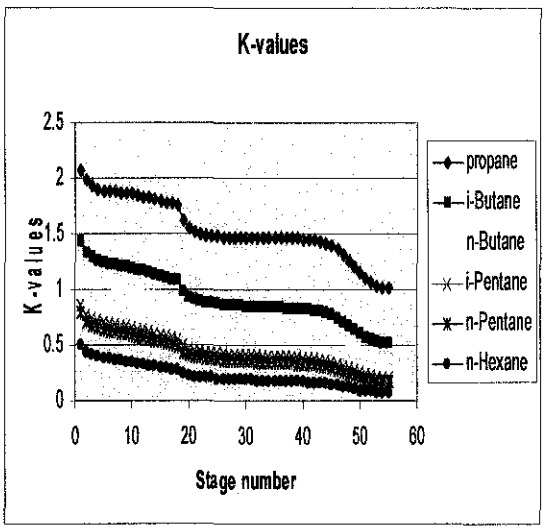


Figure 4.3.2(b): Converged distribution coefficient for each component (HYSYS)

Comparing the two graphs, it can be seen that the trend of graph obtained from MODEL is almost the same as the trend of graph from HYSYS. The difference is that the slope for HYSYS steadily decreases as the stage number increases, but the slope for MODEL dramatically decreases from the initial stage and become constant as it reached to the feed stage, which is at stage 17. Then the slope decreases back and become constant at about stage 23 towards the end of stage. Overall, it can be said that the value of equilibrium constant is proportional to temperature.

The possibility of the difference between the result from HYSYS and MODEL is basically because of the assumption of K-values as the approximate K-values. For the ideal system, the vapor and liquid phases are assumed to be ideal vapor and liquid mixture respectively. The distribution constant, K depends only on temperature and pressure. This is an approximate K values because most of the system are approximately independent of composition. The equilibrium calculations were

performed by using the McWilliams regression equation, which fitted the DePriester charts that is normally used for equilibrium calculations. But for the non-ideal system, the K-values depend on pressure, temperature, and composition of components. However due to the time constraints, the non-ideal K-values cannot be determined. So, the initial K values are used in the modeling, which explains on why the trend of K values obtained from Model is slightly differs from the K values obtained from HYSYS.

4.3.3 Temperature Profile

The temperature profile is obtained from the boiling point calculation and iteration until converged temperature profile is obtained.

The graphs for temperature profile from MODEL which is Figure 4.3.3(a) and HYSYS which is Figure 4.3.3(b) are shown below. The curve of temperature profile for MODEL is not as smooth as the curve of the temperature profile for HYSYS. The temperature of reboiler and condenser for HYSYS is 141°C and 42.38°C respectively. This value is somehow different with the value obtained from MODEL which is 150.79°C for reboiler and 38.76°C for the condenser.

From the Figure 4.3.3(a) and Figure 4.3.3(b) shown below, it can be seen that the temperature decreases as the stage number increases which makes sense because the lowest stage number is the reboiler which will basically generate more heat to vaporize the liquid and the highest stage number is the condenser which this is the stage where condensation occurs so temperature would be much smaller at the condenser. There are small change from the stage number 10 to 16 which is at the stripping section and stage 24 to the end of stage which is the rectifying section. This can be explained by the possibility of improper placing of feed stage at the distillation column. Improper placing of feed stage will basically makes the stage to be over-staged. For example, if the feed is located in the stripping section, there will be less stripping section, thus results to the rectifying to be over-staged and vice versa. The effect of over-staged is that it will results to small change on the parameters such as temperatures along the column.

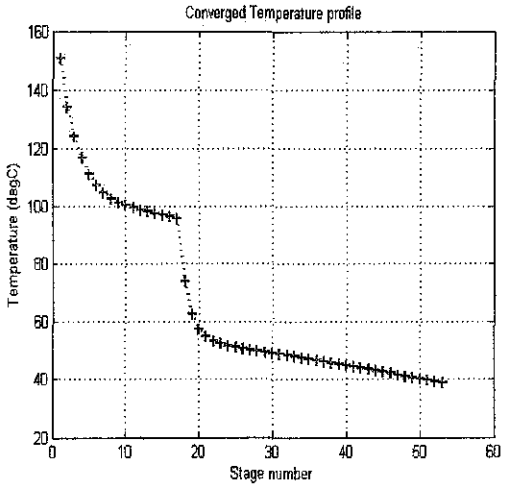


Figure 4.3.3(a) Temperature profile (MODEL)

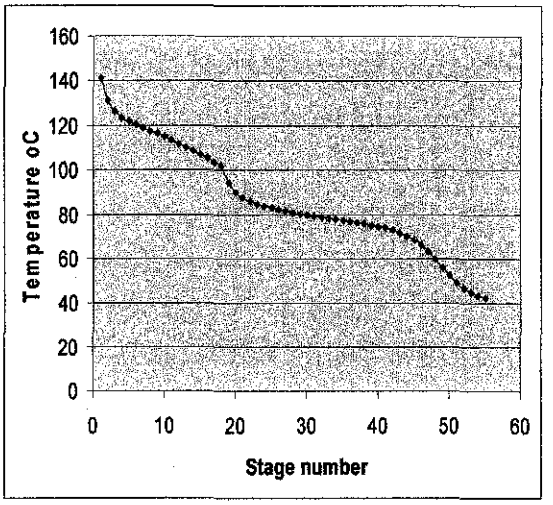


Figure 4.3.3 (b): Temperature profile (HYSYS)

4.3.4 Condenser and Reboiler Duty

The function of reboiler is basically to boil more liquids as the liquids flows downwards to the column so that the liquid will vaporized again and flows upwards to the column. The reboiler duty is important so that there will be no excess of heating in the column otherwise it will affect the top product purity. On the other hand, the condenser tends to condense the vapor that flows upwards the column to be liquid so that it will flow back downwards the column. In the model, the reboiler duty for ideal system is 1.3571×10^7kJ/hr and 1.08784×10^7 kJ/hr for non-ideal system. The condenser duty for ideal system in MODEL is -1.0956×10^7 kJ/hr and -0.85083×10^7 kJ/hr for non ideal system. Whereas the value of reboiler and condenser duty obtained from HYSYS are 1.0011×10^7 kJ/hr and 1.3151×10^7 kJ/hr respectively. Comparing the results between ideal and non-ideal system, the reboiler and condenser duty for non-ideal system have closer and more precise results compared to the values obtained from HYSYS.

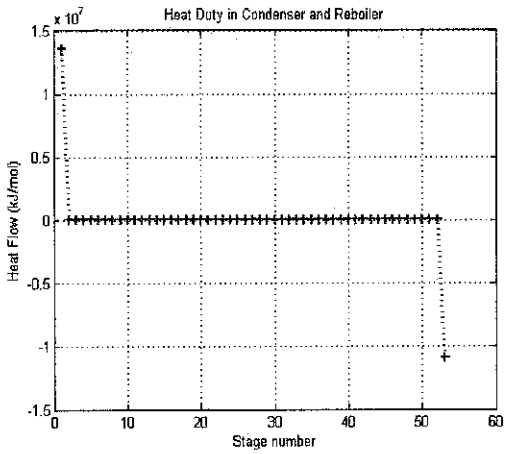


Figure 4.3.4(a): Ideal Reboiler and Condenser Heat Duty

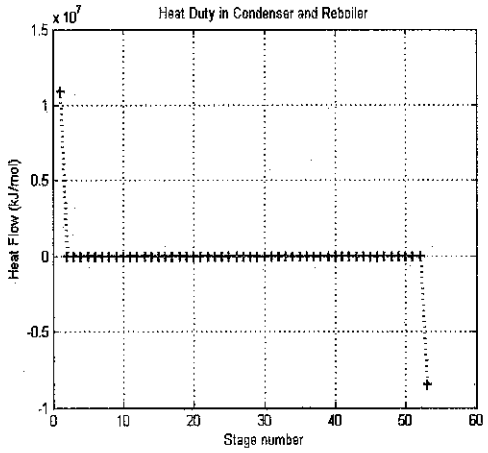


Figure 4.3.4(b): Non-Ideal Reboiler and Condenser Heat Duty

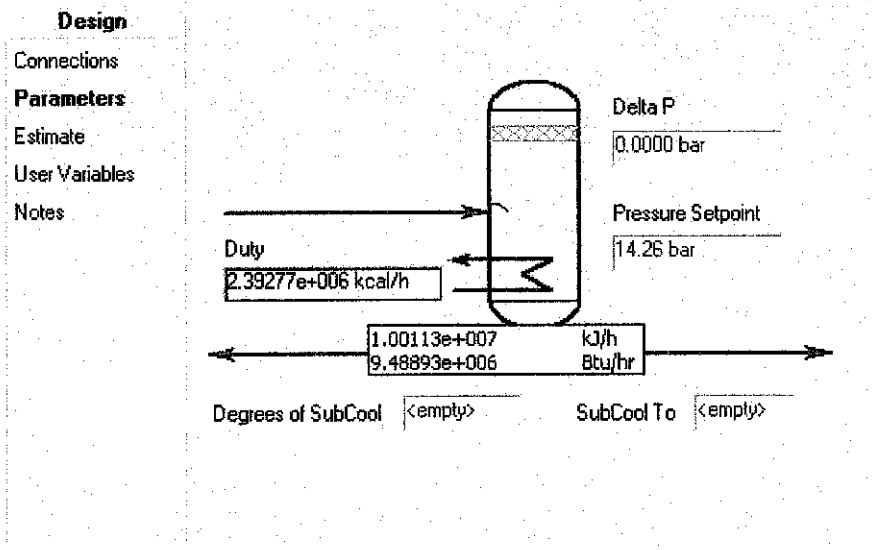


Figure 4.3.4(c): Reboiler Heat Duty (HYSYS)

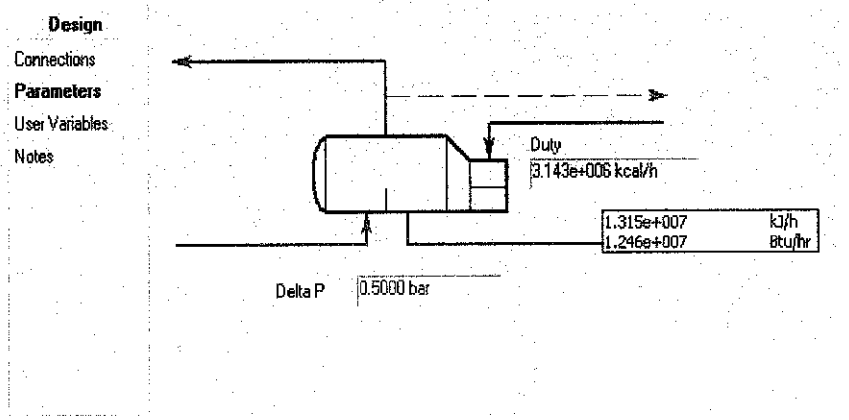


Figure 4.3.4(d): Condenser Duty (HYSYS)

Table 4.3.4 : Values of reboiler and condenser duty from HYSYS, MODEL with ideal mixture assumption and MODEL with non-ideal mixture assumption

	Reboiler (kJ/h)	Condenser
HYSYS	1.0011x107	1.3151x107
MODEL (ideal)	1.3571x107	-1.0956x107
MODEL (non ideal)	1.08784x107	-0.85083x107

4.4 Results of energy balance

4.4.1 Ideal mixtures vapor molar enthalpy

The vapor molar enthalpy is computed from EQ11, by integrating for each components, an equation in zero-pressure heat capacity at constant pressure, C^{o}_{pv}, starting from reference temperature T^{o}, to the temperature of interest. Typically To is taken as 0K or 25°C. For an ideal gas, the enthalpy is independent of pressure, meaning that the pressure has no effect to the enthalpy.

$$C^{\circ}p_v = a_1 + a_2T + a_3(T^2) + a_4(T^3) \quad \text{----------(EQ11)}$$

where ak depends on the species. Values for constants for hundreds of compounds are tabulated by Reid, Prausnitz, and Poling. The enthalpy of vapor molar enthalpy can be calculated from

$$h^{\circ}_{v} = \int_{To}^{T} C^{\circ}p_v dT = \sum_{k=1}^{4} \frac{a_k(T_k - T^{\circ}k)}{k} \quad \text{----------(EQ12)}$$

4.4.2 Ideal Mixtures liquid molar enthalpy

The enthalpy for ideal liquid mixture is obtained by subtracting the molar enthalpy of vaporization from the ideal vapor molar enthalpy for each species, as given in equation, and summarizing these as shown by EQ13

$$h_L = \sum_{i=1}^{C} x_i (h^\circ_{iv} - \Delta h_{i,vap}) \quad \text{---------- (EQ13)}$$

4.4.3 Real Liquid Mixtures

The value for real liquid mixtures can be obtained by using the equation below:

$$H_l = \sum_{i=1}^{\ell} x_i H_i + \Delta H_{mix} \quad \text{---------- (EQ14)}$$

where ΔH_{mix} is the *enthalpy change of mixing*

$$\Delta H_{mix} = H(T,P,x) - x_1 H_1(T,P) - x_2 H_2(T,P) .... - x_n H_n(T,P) \quad \text{-- (EQ15)}$$

For liquid mixtures, the value of enthalpy change of mixing is equals to the value of excess mixing property, H^{ex}.

$$H^{ex} = \Delta H_{mix} \quad \text{----------(EQ16)}$$

$$H^{ex} = \sum x_i (H_i - H_i^{IM}) \quad \text{----------(EQ17)}$$

The superscript IM indicates the ideal mixture properties. The value of Hex can be found from the equation

$$G^{ex} = H^{ex} - TS^{ex} \quad \text{----------(EQ18)}$$

For the sake of simplicity of the calculation, it is assumed that the mixtures are free from strong associates forces such as hydrogen bonding between each other, resulting the value for S^{ex} is zero for all components. Thus,

$G^{ex} = H^{ex} = \Delta H_{mix}$ ---------- (EQ19)

So EQ14 becomes

$$H_l = \sum_{i=1}^{\ell} x_i H_i + G_{ex}$$ ---------- (EQ20)

The value for excess Gibbs free energy can be calculated from the equation

$G^{ex} = RT \sum x_i \ln \gamma_i$ ----------(EQ21)

γi is the activity coefficient and can be calculated by using the equation

$$\gamma_i = \frac{f_i^{\circ}}{f_i x_i}$$ ----------(EQ22)

fio is the pure fugacity of component i and f_i is the fugacity of component i in liquid mixtures.

4.4.4 Real Gaseous Mixtures

The method of calculating the enthalpy of real gas mixture is almost the same with the calculation for the real liquid mixtures; the only difference is that the calculation will use the component mole fraction for vapor phase, fugacities of each component at gas phase and thus

$$H_v = \sum_{i=1}^{\ell} x_i H_i + G^{ex}$$ ---------- (EQ23)

where the value of $\sum_{i=1}^{\ell} x_i H_i$ is the ideal vapor molar enthalpy which is obtained earlier.

The energy balance calculations were incorporated into the constant molal overflow. The generated energy balance matrix is singular. In order to achieve the inverse of the singular matrix, singular value decomposition (SVD) is applied in the model. The SVD will diagnose precisely the problem and solved it especially when the cases where the Gaussian elimination and LU decomposition fail to give satisfactory results.

The results for enthalpy for ideal and real mixtures of liquid and vapor enthalpy at each stage number are shown below.

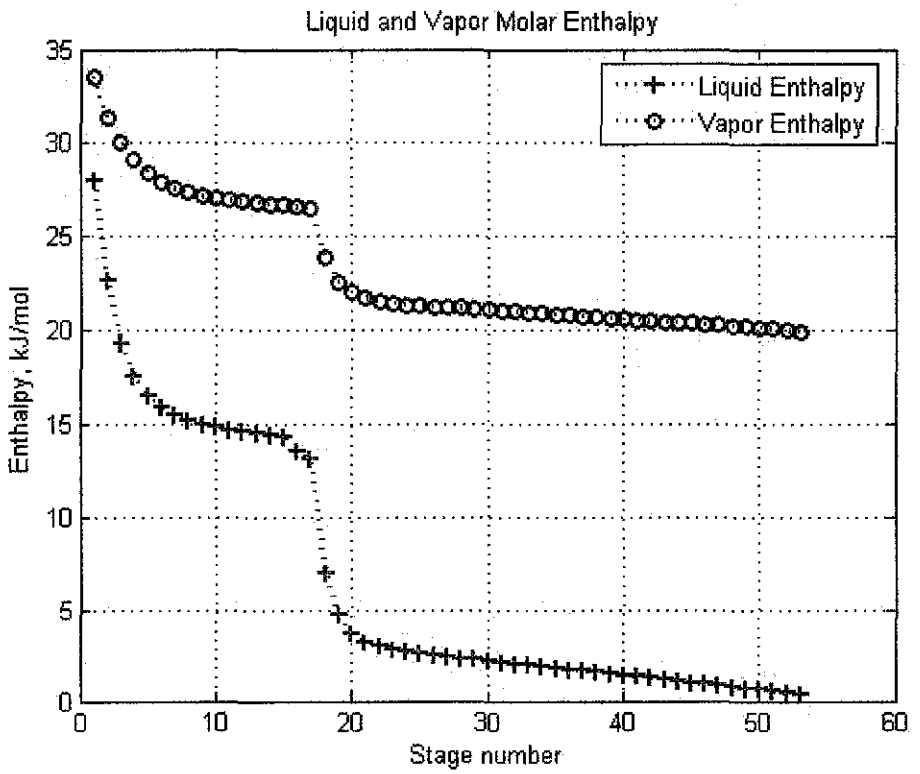


Figure 4.4.4(a): Liquid and Vapor Molar enthalpy for Ideal Mixture

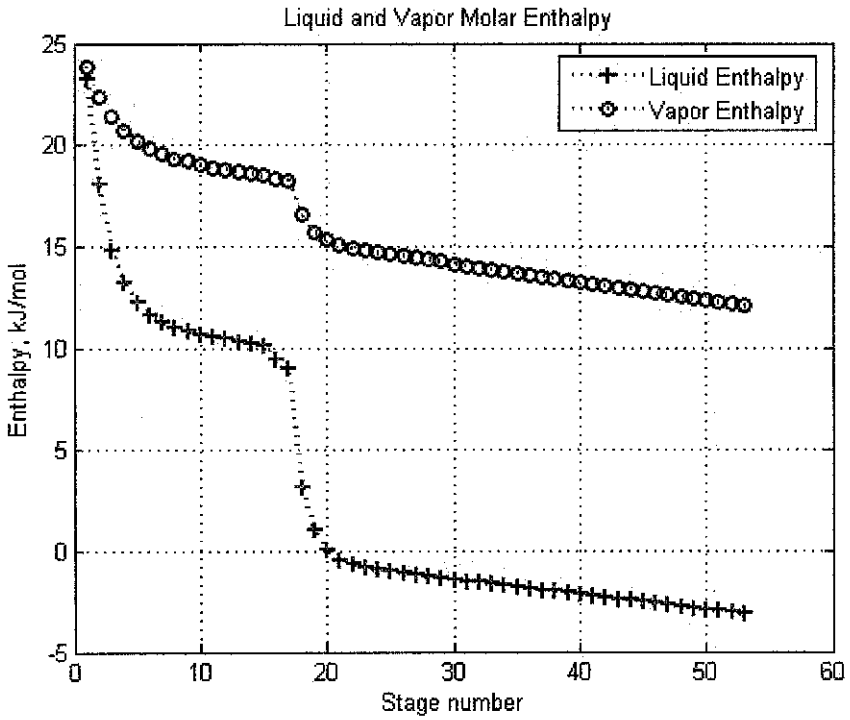


Figure 4.4.4(b): Liquid and Vapor for Non-Ideal Mixture

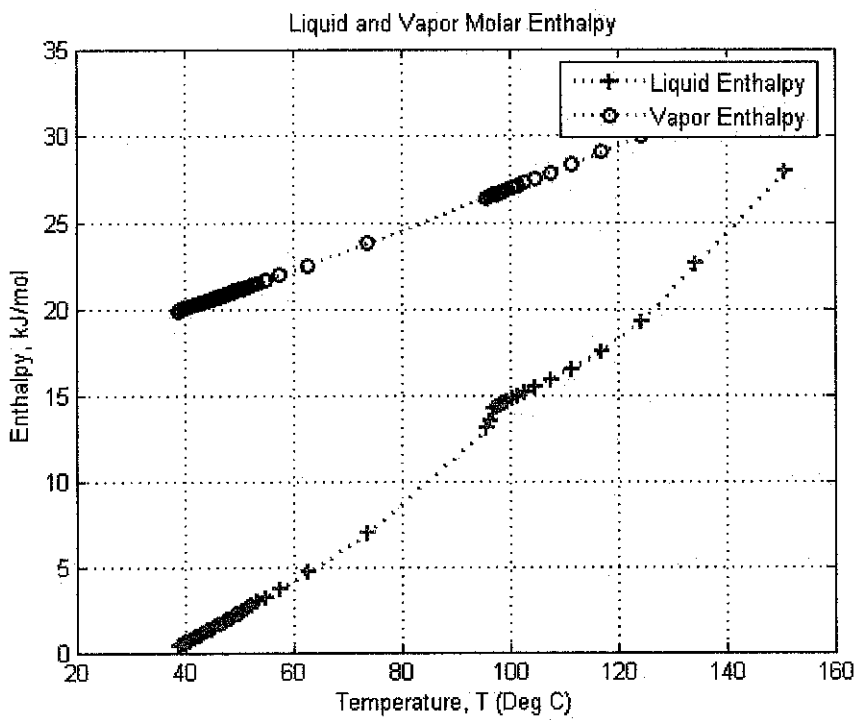


Figure 4.4.4(c): Enthalpy versus Temperature for Ideal Mixture

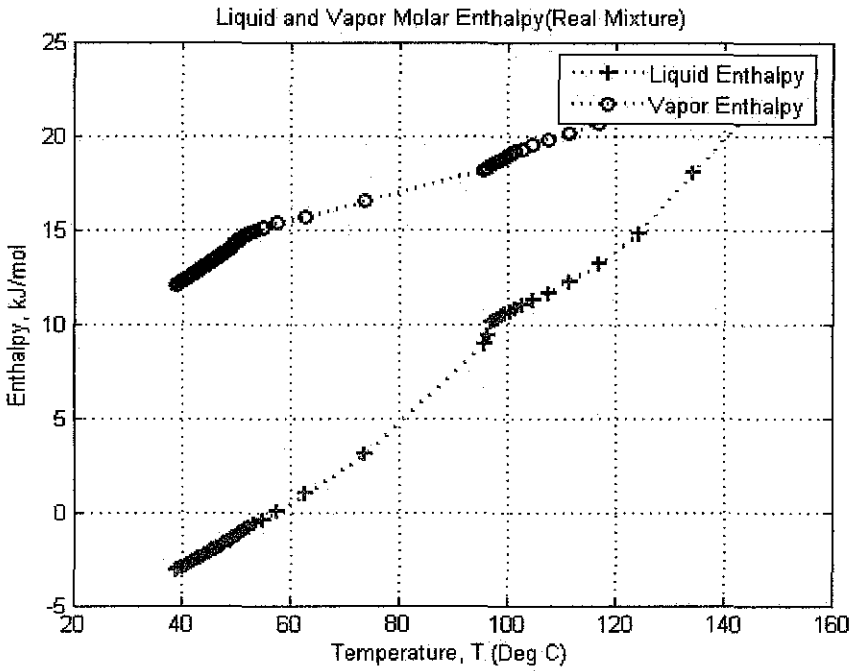


Figure 4.4.4(d): Enthalpy versus Temperature Non-Ideal Mixture

From the figures above, it is observed that the value of molar enthalpy for ideal and non ideal mixtures increases as the temperature increases. The initial value for both vapor and liquid molar enthalpy for real mixtures are basically smaller compared to the initial value for vapor and liquid molar enthalpy of ideal mixtures. As the temperature increases, the values for real mixtures enthalpy increases more than the ideal mixtures enthalpy. The difference between the initial and final value for vapor and liquid molar enthalpy for real mixture is smaller compared to the one for ideal mixtures. The pattern of graph shows that the slope for non ideal mixture is steeper than the slope of enthalpies for ideal mixtures, especially for the non ideal liquid molar enthalpy at high temperature.

Overall both ideal and non ideal mixtures enthalpy for vapor and liquid are proportional to the temperature. This can be explained from the ideal mixture calculations where the liquid molar enthalpy was based from the vapor molar enthalpy and the heat of vaporization, both of which are functions of temperature. Same goes to the non ideal mixtures where the calculation for liquid and vapor molar enthalpy for non ideal mixture is based from the summation of ideal liquid and vapor molar enthalpy plus the excess of Gibbs free energy, which is also the function of

temperature. The values for the non ideal mixture is somewhat smaller compared to the ideal mixture. This can be explained from the basic theory of non ideal mixture where in non ideal mixture, the energy between molecules is taken into consideration. So, some energy has been used during the interaction resulting to the enthalpy values achieved are smaller.

4.5 Narrow boiling range for feed mixtures

Narrow feed range is used when the dew point and bubble point temperature are not far apart. In the narrow feed boiling procedure, the compositional gradient closely related to temperature gradient, due to the temperature dependence of the equilibrium phase constants. In other words, in a relatively narrow temperature range, small change in temperature will results to the large percentage of change of compositional gradient.

4.6 Error analysis

The standard error is a measure of the amount of error in the prediction of temperature profile given by the MODEL with the reference from the HYSYS. The equation used for Sum of square error (SSE) and Root mean square error (RMSE) is given by:

$$SSE = \frac{1}{N}\sum (y_i - y_i^{\circ})^2 \qquad RMSE = \frac{1}{N}\sqrt{\sum (y_i - y_i^{\circ})^2}$$

where

N = sample number (number of stages)

y_i = Parameter profile given by MODEL

y_i° = Parameter profile given by HYSYS

From the figures obtained before,

1. Temperature Profile

   SSE = 484.5941

   RMSE = 3.023798

2. Equilibrium Coefficient, K

   SSE = 0.128624

   RMSE = 0.000229

3. Propane Recovery

   SSE = 0.273204

   RMSE = 0.009862

4.7 Other assumptions made

4.7.1 Linear pressure variation

The pressure of each stage in the column is assumed to be linear variation along the stage column. This is because of lack of data of the stage to stage pressure. To ensure that this assumption is valid, the pressure drop is assumed to be constant across each tray. Thus, the K-values can be calculated from this pressure profile.

4.7.2 No C2 (Ethane) content in feed

It is also assumed that the feed which enters the Depropanizer has no ethane. This assumption was made based on the general plant design which is the ethane will be removed first in Deethanizer. So the feed will only have six components with the propane to be the lightest component in the feed mixture.

4.7.3 No C5 and C6 in distillate

During overall mass balance, there will be no component of pentane and hexane in the top product. This is made because of the available plant data obtained does not contain any pentane or hexane in the top product.

4.7.4 No hydrogen bonding in the system

Hydrogen bonding is an intermolecular forces that holds water together, and also known as the intermolecular forces at which hydrogen is bonded to a nitrogen, oxygen, or fluorine in one molecule is attracted to an unshared pair of electrons, usually on oxygen, nitrogen, or fluorine, in another molecule.

This assumption are made so that the excess entropy which is used to calculate the excess Gibbs free energy will equals to zero. This is because if the mixtures have no hydrogen bonding effect, the excess entropy for all components is zero. This is justified so that the calculation of real mixtures enthalpy can be done.

4.7.5 100% tray efficiency

The assumption of 100% tray efficiency is made because to simplify the model. This is the natural assumption to make during the building of a model of a distillation column. Thus, this assumption leads to an over design, in other words the distillation column is too efficient or have larger number of trays compared to the actual distillation column.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.0 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

In Chapter 1, the objectives of this project were stated. The objectives of this project are:

1. To model a depropanizer distillation column by removing the ideal mixture assumption using MATLAB software.

2. To develop a modeling for de-ethanizer and de-butanizer column.

3. To compare the results obtained from the model with the results from HYSYS.

4. Show the output of distillation column by using the GUI (Graphic User Interface)

The first objective is completed by the student as well as the third and fourth objective. Unfortunately the second objective cannot be completed by the student due to time constraints.

Based on the results obtained, the MODEL can be concluded to be in capable of producing results that is closed to calculated values given by HYSYS. This can be seen from the value of SSE and RMSE errors calculated in Chapter 4.

There are some results which are not the same with the results obtained from HYSYS. This is because of several such as the assumptions made for the simplification of the model. Some of them are:

1. 100% tray efficiency
2. Narrow boiling feed procedure
3. Ideal equilibrium constant

To ensure that the results from MODEL match the actual PGB plant data, it is suggested that these assumptions to be removed. Such reasons for the suggestion have been discussed in Chapter 4 and also will be explained more in Section 5.1 (Recommendations).

As a conclusion, the objectives of the final year project are almost 100% accomplished by an improvement for the modeling by incorporating the non-ideal mixtures properties into the model, and also some enhancement to the GUI such as inserting a flash simulation made by the student to the GUI by using the Macromedia Flash software.

5.2 Recommendations

Several recommendations are suggested to improve the model of this project. The recommendations are made in order to improve the performance and results obtained from the model.

5.2.1 Incorporate tray efficiency into the model

The initial assumption in this project is by assuming 100% tray efficiency of the distillation column. In order to improve the results obtained from this model, it is suggested that the assumption 100% tray efficiency to be removed and the tray efficiency procedure must be incorporated in this model.

5.2.2 Re-calculate the equilibrium constant values by using the non-ideal procedure

In this project, the K values obtained are based on the approximate K values which are only dependent on the temperature and pressure. Thus it is recommended that the K values to be re-calculated by using the non-ideal K values which dependent on the temperature, pressure and as well as the composition of each components. This is important because in distillation column, the K-values will be different from each stage not only because of the properties of temperature, pressure that change but also the components composition which also changes from stage to stage. The author did not manage to complete this calculation due to time constraints.

5.2.3 Comparing the results with PETRONAS in-house steady-state simulation software PetronasSIM

The student cannot managed to compare the results obtained with PetronasSIM because of the time constraints and difficulties to locate the simulation software in UTP. For future recommendations, it is recommended that the results are compared with the PetronasSIM as well.

5.2.4 Obtain more plant data from plant (PGB)

It is recommended to obtain more plant data from PGB such as the pressure at each stage so that the assumption of linear pressure distribution can be removed to allow more accurate results to be obtained.

5.2.5 Improvement of MODEL and HYSYS by using other Activity Coefficient Model

If there is a case where real data is not available, some most popular activity coefficient models can be used to predict the system's behavior at certain conditions. The models are the NRTL, Wilson, SRK, UNIQUAC, and UNIFAC models. These models are best used when dealing with the non-ideality of mixtures in the system.

REFERENCES

1. Perry & Green, "Perry's Chemical Engineer's Handbook" , Mc Graw Hill, Seventh Edition

2. Khirul Akmal bin Azhari, June 2004, "*Development of Distillation Column Model For PGB Depropanizer Column by Incorporating Energy Balances*", Chemical Engineering, Universiti Teknologi PETRONAS

3. Philips C. Wankat, 1998, "*Equilibrium Staged Separations*". Prentice Hall, New Jersey.

4. Irving M Klotz & Robert M Rosenberg "*Chemical Thermodynamics, Basic Theory and Methods Fifth Edition*", Wiley Interscience.

5. Stanley I. Sandler 1999"*Chemical And Engineering Thermodynamics*" Wiley.

6. Jack Winnick 1996 "*Chemical Engineering Themodynamics*", Wiley.

7. Jefferson W. Tester / Michael Modell 1996 "*Thermodynamics and its Applications*" 3rd Edition, Prentice Hall

8. J.M Smith, H. C. Van Ness, M. M. Abbott 2001 "Chemical Engineering Thermodynamics" Sixth Edition Mc Graw Hill

9. Duane Hanselman, Bruce Littlefield 1998 "*Mastering MATLAB 5 A Comprehensive Tutorial and Reference*" MATLAB Curriculum Series

APPENDICES

APPENDIX A

GRAPHICAL USER INTERFACE

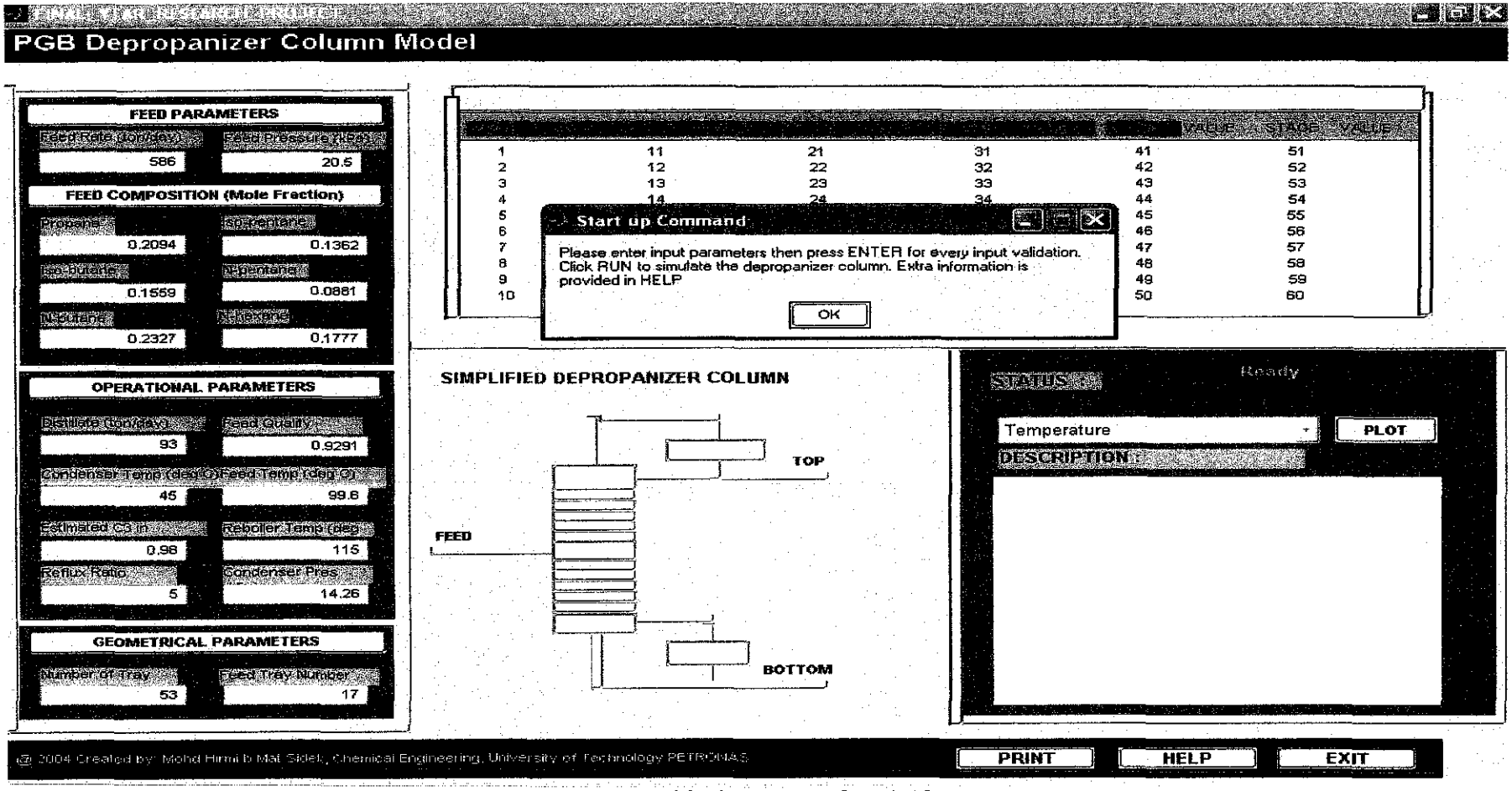


Figure 1: Graphical User Interface (MODEL)

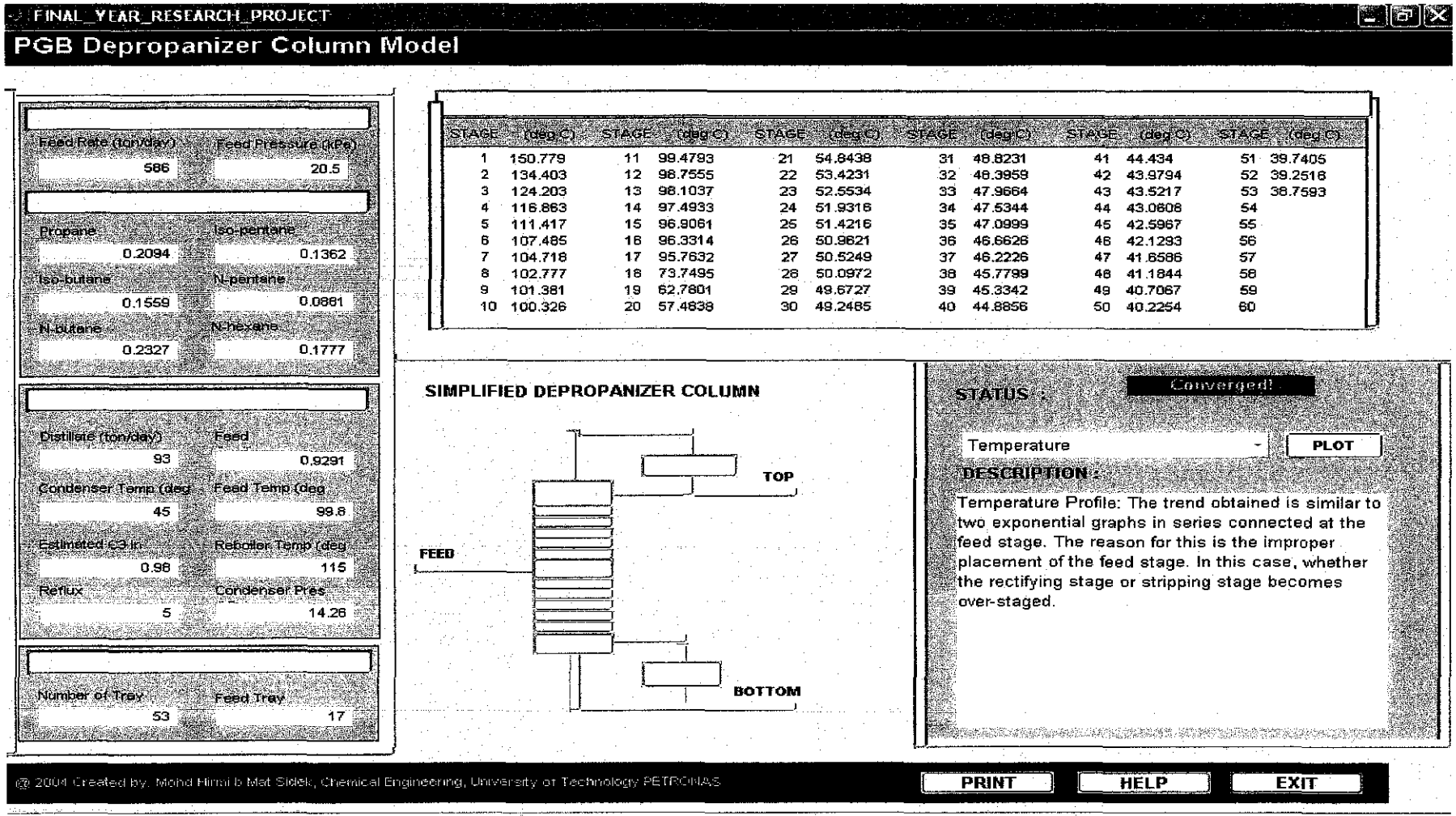


Figure 2: Graphical User Interface (MODEL)

APPENDIX B

MATLAB CODING

MATLAB CODING

```
% Title       : Model of PGB Depropanizer
% Description : This model uses the bubble point method with Theta convergence method as
outlined by Wankat (1988, pg 251). The model is a steady-state model and
is built with two underlying assumptions, which are constant molar overflow and 100%
efficiency.
%---------------------------------------------------------------------------------
% Mass Balance Codes Writer  : SHRI GUGAN A/L SIVA SUBRAMANIAM
%                    Universiti Teknologi PETRONAS
%                    Bandar Seri Iskandar
%                    Perak
% Date               : 14 NOVEMBER 2003
% Main Task             : Mass Balance


% Energy Balance Codes Writer : KHIRUL AKMAL BIN AZAHRI
%                    Universiti Teknologi PETRONAS
%                    Bandar Seri Iskandar
%                    Perak
% Date               : 8 FEBRUARY 2004
% Main Task             : Energy Balance and Graphical User Interface (GUI)

% Energy Balance Codes Writer : MOHD HIRMI B MAT SIDEK
%                    Universiti Teknologi PETRONAS
%                    Bandar Seri Iskandar
%                    Perak
% Date               : 20 OCTOBER 2004
% Main Task             : Energy Balance(Removal of Ideal Mixture Assumptions
%                    and Graphical User Interface (GUI)

%================================================================

%                 Multicomponent Distillation (6 component mixture)

%================================================================

%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ START
OF GUI @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

function varargout = FINAL_YEAR_RESEARCH_PROJECT(varargin)
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
            'gui_Singleton',  gui_Singleton, ...
            'gui_OpeningFcn', @FINAL_YEAR_RESEARCH_PROJECT_OpeningFcn, ...
            'gui_OutputFcn',  @FINAL_YEAR_RESEARCH_PROJECT_OutputFcn, ...
            'gui_LayoutFcn',  [] , ...
            'gui_Callback',   []);
if nargin & isstr(varargin{1})
   gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
   [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
```

```
else
   gui_mainfcn(gui_State, varargin{:});
end

function FINAL_YEAR_RESEARCH_PROJECT_OpeningFcn(hObject, eventdata, handles,
varargin)
handles.output = hObject;
guidata(hObject, handles);

function varargout = FINAL_YEAR_RESEARCH_PROJECT_OutputFcn(hObject,
eventdata, handles)
varargout{1} = handles.output;

msgbox('Please enter input parameters then press ENTER for every input validation. Click
RUN to simulate the depropanizer column. Extra information is provided in HELP','Start up
Command')

function edit1_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit1_Callback(hObject, eventdata, handles)
feed = str2double(get(hObject, 'String'));
if isnan(feed)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.feed = feed;
setappdata(gcbf, 'metricdata', data);

function edit2_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit2_Callback(hObject, eventdata, handles)
feedpressure = str2double(get(hObject, 'String'));
if isnan(feedpressure)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.feedpressure = feedpressure;
setappdata(gcbf, 'metricdata', data);

function edit3_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
```

```
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit3_Callback(hObject, eventdata, handles)
function pushbutton2_Callback(hObject, eventdata, handles)
function pushbutton3_Callback(hObject, eventdata, handles)
function edit5_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit5_Callback(hObject, eventdata, handles)
propane = str2double(get(hObject, 'String'));
if isnan(propane)
    set(hObject, 'String', 0);
    errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.propane = propane;
setappdata(gcbf, 'metricdata', data);

function edit6_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit6_Callback(hObject, eventdata, handles)
function edit7_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit7_Callback(hObject, eventdata, handles)
isobutane = str2double(get(hObject, 'String'));
if isnan(isobutane)
    set(hObject, 'String', 0);
    errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.isobutane = isobutane;
setappdata(gcbf, 'metricdata', data);

function edit8_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

```
function edit8_Callback(hObject, eventdata, handles)
function edit9_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit9_Callback(hObject, eventdata, handles)
nbutane = str2double(get(hObject, 'String'));
if isnan(nbutane)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.nbutane = nbutane;
setappdata(gcbf, 'metricdata', data);

function edit10_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit10_Callback(hObject, eventdata, handles)
function pushbutton4_Callback(hObject, eventdata, handles)
function edit11_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit11_Callback(hObject, eventdata, handles)
disflow = str2double(get(hObject, 'String'));
if isnan(disflow)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.disflow = disflow;
setappdata(gcbf, 'metricdata', data);


function edit12_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit12_Callback(hObject, eventdata, handles)
feedQ = str2double(get(hObject, 'String'));
if isnan(feedQ)
```

```
    set(hObject, 'String', 0);
    errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.feedQ = feedQ;
setappdata(gcbf, 'metricdata', data);

function edit13_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit13_Callback(hObject, eventdata, handles)
conT = str2double(get(hObject, 'String'));
if isnan(conT)
    set(hObject, 'String', 0);
    errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.conT = conT;
setappdata(gcbf, 'metricdata', data);
function edit14_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit14_Callback(hObject, eventdata, handles)
feedT = str2double(get(hObject, 'String'));
if isnan(feedT)
    set(hObject, 'String', 0);
    errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.feedT = feedT;
setappdata(gcbf, 'metricdata', data);


function edit15_CreateFcn(hObject, eventdata, handles)
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit15_Callback(hObject, eventdata, handles)
estimate = str2double(get(hObject, 'String'));
if isnan(estimate)
    set(hObject, 'String', 0);
    errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
```

```
data.estimate = estimate;
setappdata(gcbf, 'metricdata', data);


function edit16_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit16_Callback(hObject, eventdata, handles)
rebT = str2double(get(hObject, 'String'));
if isnan(rebT)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.rebT = rebT;
setappdata(gcbf, 'metricdata', data);


function edit17_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit17_Callback(hObject, eventdata, handles)
refrat = str2double(get(hObject, 'String'));
if isnan(refrat)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.refrat = refrat;
setappdata(gcbf, 'metricdata', data);


function edit18_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit18_Callback(hObject, eventdata, handles)
conP = str2double(get(hObject, 'String'));
if isnan(conP)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.conP = conP;
```

```
setappdata(gcbf, 'metricdata', data);

function edit19_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit19_Callback(hObject, eventdata, handles)
isopentane = str2double(get(hObject, 'String'));
if isnan(isopentane)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.isopentane = isopentane;
setappdata(gcbf, 'metricdata', data);

function edit20_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit20_Callback(hObject, eventdata, handles)
npentane = str2double(get(hObject, 'String'));
if isnan(npentane)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.npentane = npentane;
setappdata(gcbf, 'metricdata', data);

function edit21_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit21_Callback(hObject, eventdata, handles)
nhexane = str2double(get(hObject, 'String'));
if isnan(nhexane)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.nhexane = nhexane;
setappdata(gcbf, 'metricdata', data);

function figure1_ResizeFcn(hObject, eventdata, handles)
function pushbutton5_Callback(hObject, eventdata, handles)
```

```
function pushbutton6_Callback(hObject, eventdata, handles)
function pushbutton7_Callback(hObject, eventdata, handles)
function pushbutton8_Callback(hObject, eventdata, handles)
function pushbutton9_Callback(hObject, eventdata, handles)
function pushbutton10_Callback(hObject, eventdata, handles)
function pushbutton11_Callback(hObject, eventdata, handles)
function pushbutton12_Callback(hObject, eventdata, handles)
function pushbutton13_Callback(hObject, eventdata, handles)
function pushbutton14_Callback(hObject, eventdata, handles)
function pushbutton15_Callback(hObject, eventdata, handles)
printdlg
function pushbutton16_Callback(hObject, eventdata, handles)
  stat = web ('d:\MATLAB-files\HelpBook\HelpBook.html', '-browser')
function pushbutton17_Callback(hObject, eventdata, handles)
close
function pushbutton18_Callback(hObject, eventdata, handles)
function pushbutton19_Callback(hObject, eventdata, handles)
function pushbutton20_Callback(hObject, eventdata, handles)
function pushbutton21_Callback(hObject, eventdata, handles)
function pushbutton22_Callback(hObject, eventdata, handles)
function pushbutton23_Callback(hObject, eventdata, handles)
function pushbutton24_Callback(hObject, eventdata, handles)
function pushbutton25_Callback(hObject, eventdata, handles)
function edit22_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit22_Callback(hObject, eventdata, handles)
numtray = str2double(get(hObject, 'String'));
if isnan(numtray)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
data.numtray = numtray;
setappdata(gcbf, 'metricdata', data);


function edit23_CreateFcn(hObject, eventdata, handles)
if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit23_Callback(hObject, eventdata, handles)
fnum = str2double(get(hObject, 'String'));
if isnan(fnum)
   set(hObject, 'String', 0);
   errordlg('Input must be a number','Error');
end
data = getappdata(gcbf, 'metricdata');
```

```
data.fnum = fnum;
setappdata(gcbf, 'metricdata', data);


%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ END
OF GUI @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@


%##########################################################################
function pushbutton111_Callback(hObject, eventdata, handles,Feed)
data = getappdata(gcbf, 'metricdata');
%##########################################################################

F = (data.feed)*1000;
za = data.propane;
zb = data.isobutane;
zc = data.nbutane;
zd = data.isopentane;
ze = data.npentane;
zf = data.nhexane;
xad= data.estimate;
Tf = data.feedT;
RR = data. refrat;
D  = (data.disflow)*1000;
N  = data.numtray;
Nf = data.fnum;
T_con = data.conT;
T_reb = data.rebT;
P_con = (data.conP)*1e5;
P_feed = (data.feedpressure)*1e5;
q = data.feedQ;


%##########################################################################%-------
----------------------OVERALL MASS BALANCE----------------------------------
%##########################################################################

% Bottoms flowrate, B
% Liquid flow in column in Rectifying section, Lr
% Liquid flow in column in Stripping section,    Ls
% Liquid flow at each stage for Stripping section, L(j)
% Liquid flow at each stage for Rectifying section, L(k)
% Vapor flow in column in Rectifying section, Vr
% Vapor flow in column in Stripping section, Vs
% Vapor flow at each stage for Stripping section, V(j)
% Vapor flow at each stage for Rectifying section, V(K)

B = F - D; Lr = RR * D; Ls = Lr + q * F; L(1) = B; nn = [1:N]; z = [za zb zc zd ze zf];

for j = 2 : Nf;
   L(j) = Ls;
end

for k = (Nf + 1) : N;
```

```
   L(k) = Lr;
end

Vr = Lr + D; Vs = Vr - (1 - q) * F;

for j = 1 : (Nf-1);
   V(j) = Vs;
end

for k = Nf : (N-1);
   V(k) = Vr;
end

V(N) = 0;

% Calculate Distillate component compositions, xid
% Assuming equal distribution of isomers in distillation and bottoms

xbd = (zb / (zc + zb)) * (1 - xad); xcd = (zc / (zc + zb)) * (1 - xad);
xdd = 0; xed = 0; xfd = 0;
xd = [xad xbd xcd xdd xed xfd];

% Calculate Bottoms component compositionsx, xfd

xab = (F * za - D * xad) / B; xdb = F * zd / B; xeb = F * ze / B; xfb = F * zf / B;
xbb = (zb / (zc + zb)) * (1 - xab - xeb - xdb - xfb);
xcb = (zc / (zc + zb)) * (1 - xab - xeb - xdb - xfb);
xb = [xab xbb xcb xdb xeb xfb];

% Calculate stages Temperature using linear variation from condenser to reboiler, T(j)
% Calculate stages Pressure using linear variation from condenser to reboiler, P(j)

for j = 1 : N;
   T(j) = T_reb - ( (j - 1) / (N - 1) ) * (T_reb - T_con);
end

for j = 1 : N;
   P(j) = P_con + ((j - N) / (Nf - N)) * (P_feed - P_con);
end

% Calculate K-value for each component for each stage, K(j i)
% fki is a function file

for j = 1 : N;
   t = T(j); p = P(j); ka(j) = fka(t, p); kb(j) = fkb(t, p); kc(j) = fkc(t, p);
   kd(j) = fkd(t, p);  ke(j) = fke(t, p); kf(j) = fkf(t, p);
end
   K = [ka' kb' kc' kd' ke' kf'];

%-------------------------------------------------------------------------------
COMPONENT MASS BALANCE
%-------------------------------------------------------------------------------

% Perform overall loop from mass to energy balance
convergence = 2;
```

```
counter2=0;
while abs(convergence) > 1e-3;

% Perform inner loop for temperature profile convegence
counter = 0;
theta = 10;
del_theta = 1;
while abs(del_theta) > 1e-5;

% Calculate values of elements for matrix ABC, and D

  for j = 1;
  Ba(j) = 1 + (V(j) * ka(j)) / B ; Ca(j) = -1; Da(j) = 0; Bb(j) = 1 + (V(j) * kb(j)) / B;
  Cb(j) = -1; Db(j) = 0; Bc(j) = 1 + (V(j) * kc(j)) / B; Cc(j) = -1; Dc(j) = 0;
  Bd(j) = 1 + (V(j) * kd(j)) / B; Cd(j) = -1; Dd(j) = 0; Be(j) = 1 + (V(j) * ke(j)) / B;
  Ce(j) = -1; De(j) = 0; Bf(j) = 1 + (V(j) * kf(j)) / B; Cf(j) = -1; Df(j) = 0;
  end


  for j = 2 : (Nf-1);
  Aa(j) = -((ka(j-1) * V(j-1)) / L(j-1)); Ba(j) = 1 + (V(j) * ka(j)) / L(j); Ca(j) = -1;
  Da(j) = 0; Ab(j) = -((kb(j-1) * V(j-1)) / L(j-1)); Bb(j) = 1 + (V(j) * kb(j)) / L(j) ;
  Cb(j) = -1; Db(j) = 0; Ac(j) = -((kc(j-1) * V(j-1)) / L(j-1));
  Bc(j) = 1 + (V(j) * kc(j)) / L(j); Cc(j) = -1; Dc(j) = 0;
  Ad(j) = -((kd(j-1) * V(j-1)) / L(j-1)); Bd(j) = 1 + (V(j) * kd(j)) / L(j); Cd(j) = -1;
  Dd(j) = 0; Ae(j) = -((ke(j-1) * V(j-1)) / L(j-1)); Be(j) = 1 + (V(j) * ke(j)) / L(j) ;
  Ce(j) = -1; De(j) = 0; Af(j) = -((kf(j-1) * V(j-1)) / L(j-1));
  Bf(j) = 1 + (V(j) * kf(j)) / L(j); Cf(j) = -1; Df(j) = 0;
  end

  for j = Nf;
  Aa(j) = -((ka(j-1) * V(j-1)) / L(j-1)); Ba(j) = 1 + (V(j) * ka(j)) / L(j); Ca(j) = -1;
  Da(j) = F*za; Ab(j) = -((kb(j-1) * V(j-1)) / L(j-1)); Bb(j) = 1+(V(j) * kb(j)) / L(j);
  Cb(j) = -1; Db(j)=F*zb; Ac(j) = -((kc(j-1)*V(j-1))/L(j-1)); Bc(j)=1+(V(j)*kc(j))/L(j);
  Cc(j) = -1; Dc(j)=F*zc; Ad(j) = -((kd(j-1)*V(j-1))/L(j-1)); Bd(j)=1+(V(j)*kd(j))/L(j);
  Cd(j) = -1; Dd(j)=F*zd; Ae(j) = -((ke(j-1)*V(j-1))/L(j-1)); Be(j)=1+(V(j)*ke(j))/L(j);
  Ce(j) = -1; De(j)=F*ze; Af(j) = -((kf(j-1)*V(j-1))/L(j-1)); Bf(j)=1+(V(j)*kf(j))/L(j);
  Cf(j) = -1; Df(j)=F*zf;
  end

  for j = (Nf+1) : (N-1);
  Aa(j) = -((ka(j-1) * V(j-1)) / L(j-1)); Ba(j) = 1 + (V(j) * ka(j)) / L(j); Ca(j) = -1;
  Da(j) = 0;
  Ab(j) = -((kb(j-1) * V(j-1)) / L(j-1)); Bb(j) = 1 + (V(j) * kb(j)) / L(j); Cb(j) = -1;
  Db(j) = 0;
  Ac(j) = -((kc(j-1) * V(j-1)) / L(j-1)); Bc(j) = 1 + (V(j) * kc(j)) / L(j); Cc(j) = -1;
  Dc(j) = 0;
  Ad(j) = -((kd(j-1) * V(j-1)) / L(j-1)); Bd(j) = 1 + (V(j) * kd(j)) / L(j); Cd(j) = -1;
  Dd(j) = 0;
  Ae(j) = -((ke(j-1) * V(j-1)) / L(j-1)); Be(j) = 1 + (V(j) * ke(j)) / L(j); Ce(j) = -1;
  De(j) = 0;
  Af(j) = -((kf(j-1) * V(j-1)) / L(j-1)); Bf(j) = 1 + (V(j) * kf(j)) / L(j); Cf(j) = -1;
  Df(j) = 0;
  end
```

```
for j = N;
Aa(j) = -((ka(j-1) * V(j-1)) / L(j-1)); Ba(j) = 1 + D / L(j); Da(j) = 0;
Ab(j) = -((kb(j-1) * V(j-1)) / L(j-1)); Bb(j) = 1 + D / L(j); Db(j) = 0;
Ac(j) = -((kc(j-1) * V(j-1)) / L(j-1)); Bc(j) = 1 + D / L(j); Dc(j) = 0;
Ad(j) = -((kd(j-1) * V(j-1)) / L(j-1)); Bd(j) = 1 + D / L(j); Dd(j) = 0;
Ae(j) = -((ke(j-1) * V(j-1)) / L(j-1)); Be(j) = 1 + D / L(j); De(j) = 0;
Af(j) = -((kf(j-1) * V(j-1)) / L(j-1)); Bf(j) = 1 + D / L(j); Df(j) = 0;
end

% Build ABC matrix and set all elements equals to zero

 ABCa = zeros(N); ABCb = zeros(N); ABCc = zeros(N); ABCd = zeros(N); ABCe =
zeros(N);
 ABCf = zeros(N);

% Insert values for ABC matrix

for j = 1;
ABCa(j,j) = Ba(j); ABCa(j,(j+1)) = Ca(j); ABCb(j,j) = Bb(j); ABCb(j,(j+1)) = Cb(j);
ABCc(j,j) = Bc(j); ABCc(j,(j+1)) = Cc(j); ABCd(j,j) = Bd(j); ABCd(j,(j+1)) = Cd(j);
ABCe(j,j) = Be(j); ABCe(j,(j+1)) = Ce(j); ABCf(j,j) = Bf(j); ABCf(j,(j+1)) = Cf(j);
end

for j = 2 : (N-1);
ABCa(j,j-1) = Aa(j); ABCa(j,j) = Ba(j);  ABCa(j,(j+1)) = Ca(j); ABCb(j,j-1) = Ab(j);
ABCb(j,j) = Bb(j) ; ABCb(j,(j+1)) = Cb(j);  ABCc(j,j-1) = Ac(j); ABCc(j,j) = Bc(j);
ABCc(j,(j+1)) = Cc(j); ABCd(j,j-1) = Ad(j); ABCd(j,j) = Bd(j); ABCd(j,(j+1)) = Cd(j);
ABCe(j,j-1) = Ae(j); ABCe(j,j) = Be(j);  ABCe(j,(j+1)) = Ce(j); ABCf(j,j-1) = Af(j);
ABCf(j,j) = Bf(j); ABCf(j,(j+1)) = Cf(j);
end


for j = N;
ABCa(j,j-1) = Aa(j); ABCa(j,j) = Ba(j); ABCb(j,j-1) = Ab(j); ABCb(j,j) = Bb(j);
ABCc(j,j-1) = Ac(j); ABCc(j,j) = Bc(j); ABCd(j,j-1) = Ad(j); ABCd(j,j) = Bd(j);
ABCe(j,j-1) = Ae(j); ABCe(j,j) = Be(j); ABCf(j,j-1) = Af(j); ABCf(j,j) = Bf(j);
end

% Matrix DA, DB, DC, DD, DE, DF

for j = 1:N;
DA(j,1) = Da(j); DB(j,1) = Db(j); DC(j,1) = Dc(j); DD(j,1) = Dd(j); DE(j,1) = De(j);
DF(j,1) = Df(j);
end

% Matrix D has also been build automatically
% Solve for component flow rates in matrix l, li
% Given that ABC * l = D

la = inv(ABCa) * DA; lb = inv(ABCb) * DB; lc = inv(ABCc) * DC; ld = inv(ABCd) * DD;
le = inv(ABCe) * DE; lf = inv(ABCf) * DF;
l  = [la  lb  lc  ld  le lf];

%-------------------------------------------------------------------------
 THETA CONVERGENCE METHOD
```

```
%-------------------------------------------------------------------------

% The value of theta is calculated using Newton Raphson method. fcn1 and fcn2 are function
% files and are the numerator and denominator of the Newton Raphson equation respectively.
% Input for fcn1 and fcn 2 is taken from the component mass balance.

 lin1 = [l(1,1) l(1,2) l(1,3) l(1,4) l(1,5) l(1,6)];
 lin2 = [l(N,1) l(N,2) l(N,3) l(N,4) l(N,5) l(N,6)];

% The theta convergence loop is performed to calculate value for theta

 ftheta = 1;
 counter3 = 0;
 while abs(ftheta) > 1e-5;
   theta = theta + fcn1(theta, lin1, lin2, RR, F,z,D) / fcn2(theta, lin1, lin2, RR,F,z);
   if theta < -1e-5
     theta = 1;
   else
   ftheta = fcn1(theta, lin1, lin2, RR, F,z,D);
   counter3 = counter3 + 1;
   end
 end
 del_theta = abs(1 - theta);

% Determine value for testing while loop
% Calculate corrected component mole flow rates
% The sum of component flow rates at top, SumD, and bottom, SumB of column should
match the
% overall mass balance values
% The top and bottom component flow rates are corrected using the value of theta

SumD = 0; SumB = 0;

 for c = 1 : 6;
    Bxibot_calc(c) = l(1,c);
    Dxidist_calc(c) = l(N,c) / RR;
    Dxidist_cor(c) = (F * z(c)) / (1 + theta * (Bxibot_calc(c) / Dxidist_calc(c)));
    Bxibot_cor(c)  = Dxidist_cor(c) * theta * (Bxibot_calc(c) / Dxidist_calc(c));

    SumD = SumD + Dxidist_cor(c);  SumB = SumB + Bxibot_cor(c);
    rm = Dxidist_cor ./ Dxidist_calc;  sm = Bxibot_cor ./ Bxibot_calc;
 end

 for c = 1 : 6;
    for n = (Nf+1) : N;
       l_cor(n,c) = rm(c) * l(n,c);
    end

    for n = 1 : Nf;
       l_cor(n,c) = sm(c) * l(n,c);
    end
 end

 for n = 1 : N;
    sum_l_cor(n) = 0;
```

```
    for c = 1 : 6;
      sum_l_cor(n) = sum_l_cor(n) + l_cor(n,c);
    end
  end

  for j=1:N;
    L_cor1(j,1) = l_cor(j,1); L_cor2(j,1) = l_cor(j,2); L_cor3(j,1) = l_cor(j,3);
    L_cor4(j,1) = l_cor(j,4); L_cor5(j,1) = l_cor(j,5); L_cor6(j,1) = l_cor(j,6);
  end

  L_cor =[L_cor1 L_cor2 L_cor3 L_cor4 L_cor5 L_cor6];

  for j=1:N;
    L_cor2(j) = L_cor1(j,1)+L_cor2(j,1)+L_cor3(j,1)+L_cor4(j,1)+L_cor5(j,1)+L_cor6(j,1);
  end

% The component flow rates in the rectifying section are corrected using the ratio of
% corrected to uncorrected top component flow
% The component flow rates in the stripping section are corrected using the ratio of
% corrected to uncorrected bottoms component flow
% Calculate corrected mole fractions for all components at each stage

  for n = 1 : N;
    sumx(n) = 0;
    for c = 1 : 6;
      x(n,c) = l_cor(n,c) / sum_l_cor(n);
      sumx(n) = sumx(n) + x(n,c);
    end
  end

% Calculate new temperatures for each stage
% Bubble point calculation is performed using trial and error until temperature value
% converges. fcnT is a function file that is the reverse of function file fka

  for j = 1 : N;
    Kj = [ka(j) kb(j) kc(j) kd(j) ke(j) kf(j)];
    delK = 1.0;
    while abs(delK) > 1e-5;
        for c = 1 : 6;
          Kixi(c) = Kj(c) .* x(j,c);
        end
        ka_new(j) = ka(j) / sum(Kixi);
        ka_fcninput = ka_new(j);
        p = P(j);
        T_new(j) = fcnT(ka_fcninput,p);
        t = T_new(j);
        kb(j) = fkb(t, p);
        kc(j) = fkc(t, p);
        kd(j) = fkd(t, p);
        ke(j) = fke(t, p);
        kf(j) = fkf(t, p);
        Kj = [ka_new(j) kb(j) kc(j) kd(j) ke(j) kf(j)];

        for c = 1 : 6;
          K_new(j,c) = Kj(c);
```

```
        end
        delK = ka_new(j) - ka(j);
        ka(j) = ka_new(j);
    end
  end

% Calculate vapor fraction of each component for each stage
        y = K_new .* x;
end
% End of Constant molar calculation, exit inner loop

for n = 1 : N
   x1(n) = x(n,1); x2(n) = x(n,2); x3(n) = x(n,3); x4(n) = x(n,4); x5(n) = x(n,5);
   x6(n) = x(n,6); y1(n) = y(n,1); y2(n) = y(n,2); y3(n) = y(n,3); y4(n) = y(n,4);
   y5(n) = y(n,5); y6(n) = y(n,6);
end



%#######################################################################
%-------------------------------- ENERGY BALANCES ----------------------------------
%#######################################################################

% Function for enthalpy calculation
% Calculating Cp's
% Cp1 = Heat Capacity of Methane
% For Reboiler:

% From Appendix II(Jack Winnick):
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
%             a           b           c           d ~
% Propane   = 1.212768    0.028782   -0.0000088      0 ~
% Isobutane = 1.676740    0.037849   -0.000012       0 ~
% n-butane  = 2.240853    0.036368   -0.000011       0 ~
% n-pentane = 2.974049    0.044510   -0.000014       0 ~
% n-hexane  = 3.762599    0.052548   -0.000016       0 ~
%~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

% Simplifying:

%   propane   n-butane  iso-butane  iso-pentane  n-pentane  n-hexane
a = [1.21276800  1.676740   2.240853    2.423523    2.974049   3.762599];
b = [0.02878200  0.037849   0.036368    0.046088    0.044510   0.052548];
c = [-0.0000088 -0.000012  -0.000011   -0.000015   -0.000014  -0.000016];
d = [  0          0          0           0           0          0     ];

% Where, Cp = aT + bT + cT^2 + dT^-2

%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@VAPOR
MOLAR                                                              ENTHALPY
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

AV =0; Hv_com = 0; To = 0; R = 8.314e-3;
```

```
for j=1:N;
   T(j) = T_new(j) + 273.15;
   for n=1:6;
      AV(n) = (a(n)*T(j) + (b(n)*T(j)^2)/2 + (c(n)*T(j)^3)/3 + (d(n)*T(j)^-1)/-1)*R;
      Hv_com(n) = y(1,n)*AV(n);
   end
      Hv(j) = Hv_com(1) + Hv_com(2) + Hv_com(3) + Hv_com(4) + Hv_com(5) +
Hv_com(6);
end

%Values for Critical Temperature, Critical Pressure and Assentric Factor
%      propane  i-butane  n-butane  i-pentane  n-pentane  n-hexane
Tc  = [ 369.83    408.1   425.12    460.4   469.7     507.6     ];
Pc  = [ 4.21e6   3.648e6  3.77e6   3.334e6  3.36e6   3.04e6     ];
w   = [ 0.149    0.1879   0.197    0.22224   0.251    0.304     ];

%Data for kij

%propane    = 1
%i-butane   = 2
%n-butane   = 3
%n-pentane  = 4
%n-hexane   = 5
%n-heptane  = 6

%      1   2    3    4   5     6
ccc =   [0 -0.007 0.003 0.011 0.027 0.001
      0   0    0    0   0     0
      0   0    0    0  0.017 -0.006
      0   0    0    0  0.06    0
      0   0    0    0   0     0
      0   0    0    0   0     0   ];

%@@@@@@@@@@ FUGACITY OF GASEOUS MIXTURES @@@@@@@@@

ZGG = 0;
%finding Tr and Pr
for j = 1:N;
   for i = 1:6 ,
      Tr(i) = T(j) / Tc(i);
      Pr(i) = P(j) / Pc(i);
   end
end

%finding values for bi(for single component)

for n = 1:6 ,
   big(n) = 0.07780*((R*Tc(n))/Pc(n));
end

%calculating values for ai(for single component)

for n=1:6 ,
   aig(n)=0.45724*(R^2)*(Tc(n)^2)/Pc(n);
end
```

```
%finding value of aij(combining rule)
for i = 1:6,
   for n = 1:6 ,
      ag(i,n)= ((aig(i)*aig(n))^0.5)*(1- ccc(n));
   end
end

%finding value of a and b

for n = 1:6 ,
 for i = 1:6 ,
   for j = 1:6 ,
      p(n) = y(i)*y(j)*ag(i,j);
      q(n) = y(i)*big(i);
   end
 end
      r = sum(p(n));
      s = sum(q(n));
end


%calculation for A , B ,Bi,Aij
for j = 1:N;
      AEE(j) = r*P(j)/((R*T(j))^2);
      BEE(j) = s*P(j)/ (R*T(j));
   for i = 1:6,
         Bc(i) = (big(i)*P(j)) / (R*T(j));
      for n = 1:6,
         Ac(i,n) = ag(i,n)*P(j) / ((R*T(j))^2);
      end
   end
end


%calculation for ZL value
for j = 1:N;

   pEE = [1 -(1-BEE(j)) (AEE(j)-((3*BEE(j))^2)-(2*BEE(j))) -((AEE(j)*BEE(j))-(BEE(j)^2)-
(BEE(j)^3))];
   ZEE = roots(pEE);

   for i = 1:3;
      if ZEE(i)>0.2&ZEE(i)<=1.0
      ZGG(j) = ZEE(i);
      end
   end

end

%calculation for sum xiAij

for i = 1:6;
   for n = 1:6;
```

```
      sxA(i) = y(n)*Ac(i,n);
   end
end

for j=1:N;
   for i = 1:6,
      NOA = (Bc(i) / BEE(j)) *(ZGG(j) - 1);
      NOB = log(ZGG(j) - BEE(j));
      NOC = AEE(j)/(2*(2^0.5)*BEE(j));
      NOD = ((2*sxA(i))/AEE(j))-(Bc(i)/BEE(j));
      NOE = ((ZGG(j)+(1+(2^0.5)*BEE(j))));
      NOF = ((ZGG(j)+(1-(2^0.5)*BEE(j))));

      UBI(j,i) =  NOA - NOB - (NOC * NOD * log (NOE/NOF));
      fugg(j,i) = (exp (UBI(j,i)))*(y(i)*P(j));
   end
end
%-----------------------------------------------------------------

%CALCULATION OF FUGACITY FOR PURE COMPONENT%

%Calculation of B value for second virial coefficient
for j = 1:N;
   for i = 1:6,
      Boo(i) = 0.083 - (0.422 / (Tr(i)^1.6));
      Bii(i) = 0.139 - (0.172 / (Tr(i)^4.2));
      Brr(i) = Boo(i) + (w(i)*Bii(i));
      Bvc(i) = (Brr(i)*R*Tc(i)) / (Pc(i));
   end


%Calculation for fugacity of pure component

   for i=1:6,
      foo(i) = P(j)* (exp((Bvc(i)*P(j))/(R*T(j))));
   end

%-----------------------------------------------------------------

%CALCULATION FOR ACTIVITY COEFFICENT

for i=1:6,
   ACC(i) = y(i)*fugg(j,i)/foo(i);
end

%CALCULATION OF ACT CORF & EXCESS GIBBS FREE ENERGY

for i=1:6,
   ADDD(i) = y(i)*(log(ACC(i)));
end

ADD(j) = sum(ADDD);

GEE(j) = R*T(j)*ADD(j);
end
```

```
for j = 1:N;
   HVR(j) = GEE(j) + Hv(j);
end


%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ LIQUID
MOLAR                                                          ENTHALPY
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@


% HEAT OF VAPORIZATION:
% --------------------
%   Heat of vaporization as a function of temperature for each component in use
%   The equation and data for coefficents used is from Perry's  (1997, pg 2-156)
%   The unit for the heat of vaporization is kJ/mol

% Temperature is converted from degrees Celcius to Kelvin


% Values for critical temeprature are obtained from Winnick (1997, pg 555)

Tc = [369.83 408.14 425.12 460.43 469.7 507.5];

   C1 = [ 2.92090  3.16670  3.62380  3.77000 3.91090 4.45440];
   C2 = [ 0.78237  0.38550  0.83370  0.39520 0.38681 0.39002];
   C3 = [-0.77319  0.00000 -0.82274  0.00000 0.00000 0.00000];
   C4 = [ 0.39246  0.00000  0.39613  0.00000 0.00000 0.00000];

for j=1:N;
   T(j) = T_new(j) + 273.15;
   for C = 1 : 6;
   Tr(C) = T(j) / Tc(C);% The reduced temperature, Tr for each conponent is calculated
   hvp(C) = 1e7 * C1(C) * (1-Tr(C)) ^ (C2(C) + C3(C)*Tr(C) + C4(C)*Tr(C)*Tr(C));
      if  T(j)> Tc(C);
          Hvp(j,C)=0;
      else Hvp(j,C) = 1e-6 * hvp(C); % convert J/kmol to kJ/mol
      end
   end
end

% The heat of vaporization for each component is calculated in J/kmol
% If the temperature exceeds the critical temperature of the component, the heat of
% vaporization for that component is equalts to zero.The heat of vaporization, hvap for
% each component is obtained, taking these constraints into account.


%@@@@@@@@ FUGACITY OF LIQUID MIXTURES @@@@@@@@@@

ZGL = 0;
%finding Tr and Pr

%finding values for bi(for single component)

for n = 1:6 ,
```

```
   bil(n) = 0.07780*((R*Tc(n))/Pc(n));
end

%calculating values for ai(for single component)

for n=1:6 ,
   ail(n)=0.45724*(R^2)*(Tc(n)^2)/Pc(n);
end


%finding value of aij(combining rule)
for i = 1:6,
   for n = 1:6 ,
      al(i,n)= ((ail(i)*ail(n))^0.5)*(1- ccc(n));
   end
end

%finding value of a and b

for n = 1:6 ,
 for i = 1:6 ,
   for j = 1:6 ,
      t(n) = x(i)*x(j)*al(i,j);
      u(n) = x(i)*bil(i);
   end
 end
      v = sum(t(n));
      vv = sum(u(n));
end

%calculation for A , B ,Bi,Aij
for j = 1:N;
      AEEL(j) = v*P(j)/((R*T(j))^2);
      BEEL(j) = vv*P(j)/ (R*T(j));
   for i = 1:6,
         Bcl(i) = (bil(i)*P(j)) / (R*T(j));
      for n = 1:6,
         Acl(i,n) = al(i,n)*P(j) / ((R*T(j))^2);
      end
   end
end


%calculation for ZL value
for j = 1:N;

   pEEL = [1 -(1-BEEL(j)) (AEEL(j)-((3*BEEL(j))^2)-(2*BEEL(j))) -((AEEL(j)*BEEL(j))-
(BEEL(j)^2)-(BEEL(j)^3))];
   ZEEL = roots(pEEL);

   for i = 1:3;
      if ZEEL(i)>0.2&ZEEL(i)<=1.0
      ZGL(j) = ZEEL(i);
      end
   end
```

```
end


%calculation for sum xiAij

for i = 1:6;
   for n = 1:6;
      sxAL(i) = x(n)*Acl(i,n);
   end
end

for j=1:N;
   for i = 1:6,
      NOAL = (Bcl(i) / BEEL(j)) *(ZGL(j) - 1);
      NOBL = log(ZGL(j) - BEEL(j));
      NOCL = AEEL(j)/(2*(2^0.5)*BEEL(j));
      NODL = ((2*sxAL(i))/AEEL(j))-(Bcl(i)/BEEL(j));
      NOEL = ((ZGL(j)+(1+(2^0.5)*BEEL(j))));
      NOFL = ((ZGL(j)+(1-(2^0.5)*BEEL(j))));

      UBIL(j,i) =  NOAL - NOBL - (NOCL * NODL * log (NOEL/NOFL));
      fugl(j,i) = (exp (UBIL(j,i)))*(x(i)*P(j));
   end
end
%CALCULATION OF FUGACITY FOR PURE COMPONENT%

%Calculation of B value for second virial coefficient
for j = 1:N;
   for i = 1:6,
      Bool(i) = 0.083 - (0.422 / (Tr(i)^1.6));
      Biil(i) = 0.139 - (0.172 / (Tr(i)^4.2));
      Brrl(i) = Bool(i) + (w(i)*Biil(i));
      Bvcl(i) = (Brrl(i)*R*Tc(i)) / (Pc(i));
   end


%Calculation for fugacity of pure component

   for i=1:6,
      fool(i) = P(j)* (exp((Bvcl(i)*P(j))/(R*T(j))));
   end

%-------------------------------------------------------------------

%CALCULATION FOR ACTIVITY COEFFICENT

for i=1:6,
   ACCl(i) = x(i)*fugl(j,i)/fool(i);
end

%CALCULATION OF ACT CORF & EXCESS GIBBS FREE ENERGY

for i=1:6,
   ADDDL(i) = x(i)*(log(ACC(i)));
```

```
end

ADDL(j) = sum(ADDDL);

GEEL(j) = R*T(j)*ADDL(j);
end

% ------------------------------------------------------------------
% For Reboiler:
% ------------------------------------------------------------------
AL = 0; calc = 0; To = 0; R = 8.314e-3; T(j)=0;

for j=1;
   for n=1:6;
   hvap(j,n) = Hvp(1,n);
   end
   T(j) = T_new(j) + 273.15;
   for n=1:6;
      AL(n) = ((a(n)*T(j) + (b(n)*T(j)^2)/2 + (c(n)*T(j)^3)/3 + (d(n)*T(j)^-1)/-1))*R;
      calc(j,n) = AL(n)-hvap(j,n);
      Hl_reb(n) = x(j,n)*calc(j,n);
   end
   Hl(j) = Hl_reb(1) + Hl_reb(2) + Hl_reb(3) + Hl_reb(4) + Hl_reb(5) + Hl_reb(6);
   HLR(j) = GEEL(j) + Hl(j);
end

%------------------------------------------------------------------
% For general stage (1<j<N):
% ------------------------------------------------------------------

j=0; AA=0; calc=0; n=0;

for j=2:(N-1);
   T(j) = T_new(j) + 273.15;
   n=0;
   for n=1:6;
      hvap(j,n) = Hvp(j,n);
      AA(n) = (a(n)*T(j) + (b(n)*T(j)^2)/2 + (c(n)*T(j)^3)/3 + (d(n)*T(j)^-1)/-1)*R;
      calc(n) = AA(n)- hvap(j,n); Hl_gen(n) = x(j,n)*calc(n);
   end
   Hl(j) = Hl_gen(1) + Hl_gen(2) + Hl_gen(3) + Hl_gen(4) + Hl_gen(5) + Hl_gen(6);
   HLR(j) = GEEL(j) + Hl(j);
end

%------------------------------------------------------------------
% For Condenser:
%------------------------------------------------------------------
j=0; AA=0; n=0; calc =0;

for j=N;
   T(j) = T_new(j) + 273.15;
   for n=1:6;
      AA(n) = (a(n)*T(j) + (b(n)*T(j)^2)/2 + (c(n)*T(j)^3)/3 + (d(n)*T(j)^-1)/-1)*R;
      calc(n) = AA(n)- Hvp(j,n); Hl_cond(n) = x(j,n)*calc(n);
   end
```

```
   Hl(j) = Hl_cond(1) + Hl_cond(2) + Hl_cond(3) + Hl_cond(4) + Hl_cond(5) + Hl_cond(6);
   HLR(j) = GEEL(j) + Hl(j);
end

%-------------------------------------------------------------
% For FEED STREAM:
%-------------------------------------------------------------
calc = 0; AA=0;

   for n=1:6;
   Tfk = Tf + 273.15;
   AA(n) = a(n)*Tfk + (b(n)*Tfk^2)/2 + (c(n)*Tfk^3)/3 + (d(n)*Tfk^-1)/-1;
   calc(n) = AA(n)*R; Hl_feed(n) = x(j,n)*calc(n);
   end
   hF = Hl_feed(1) + Hl_feed(2) + Hl_feed(3) + Hl_feed(4) + Hl_feed(5) + Hl_feed(6);

numfee = 0;

if numfee ~= Nf
   M = Nf;
   HFR = hF + GEEL(M);
end


%=============================================================
% REBOILER AND CONDENSER DUTY CALCULATION
%=============================================================

% Calculate condenser and reboiler duty
td = T_new(N); hB = Hl(1);

   for c = 1:6;
      xin(c) = x(N,c); yin(c) = K_new(N,c) * xin(c);
   end
hD = Hl(N);
Q(N) = D * (1 + RR) * (hD - Hv(N-1));
Q(1) = D * hD + B * hB - F * hF - Q(N);



% Calculate condenser and reboiler duty for real

tdr = T_new(N); hBr = HLR(1);

   for c = 1:6;
      xin(c) = x(N,c); yin(c) = K_new(N,c) * xin(c);
   end
hDr = HLR(N);
QRR(N) = D * (1 + RR) * (hDr - HVR(N-1));
QRR(1) = D * hDr + B * hBr - F * HFR - QRR(N);



%=============================================================
% ENERGY BALANCE MATRIX
```

```
%=========================================================================

% Calculate values of coeeficients for matrix AEBE and DE
j=0; AE=0; AE(1)=0; BE(N)=0;

for j = 1; BE(j) = Hv(j) - Hl(j+1); DE(j) =  Q(j) + B * (Hl(j+1) - Hl(j)); end
for j = 2:(Nf-1); AE(j)=Hl(j)-Hv(j-1); BE(j)=Hv(j)-Hv(j+1); DE(j)=B*(Hl(j+1)-Hl(j)); end
for j = Nf; AE(j)=Hl(j)-Hv(j-1); BE(j)=Hv(j)-Hv(j+1);DE(j)= B *(Hl(j+1) - Hl(j)); end
for j =(Nf+1):(N-1);AE(j)=Hl(j)-Hv(j-1);BE(j)=Hv(j)-Hv(j+1);DE(j)=B*(Hl(j+1)-Hl(j)); end
for j = N; AE(j) = Hl(N) - Hv(N-1); DE(j) = Q(N);                              end


% Build AEBE matrix
% Set all elements for matrix ABC values = zero
% Insert values for AEBE matrix

AEBE = zeros(N); AEBE(1,1) = BE(1);
for j = 2:N;
   AEBE(j,j-1) = AE(j); AEBE(j,j) = BE(j);
end


%=========================================================================
% SINGULAR MATRIX DECOMPOSITION (SVD)
%=========================================================================

% The svd command computes the matrix singular value decomposition.

[u, w, v] = svd (AEBE);

% Extracting diagonal values from w matrix

inv_w = zeros (N);
j=0;
for j= (1:N-1);
   val(j)    = w(j,j);
   val2(j)   = 1/val(j);
   inv_w(j,j) = val2(j);
end

step1 = u' * DE;
step2 = inv_w * step1;
vap   = v * step2;     % vapor flowrate for each tray


%=========================================================================
%                              LOOP FUNCTION
%=========================================================================

j=0;
  for j=2;
  Lnew(1,j)=vap(j-1) + B;
  end
j=0;
```

```
  for j=3:Nf;
  Lnew(1,j)=vap(j-1) + L(1,j-1) - vap(j-2);
  end
j=0;
  for j=Nf+1;
  Lnew(1,j) = vap(j-1) + L(1,Nf) - vap(Nf-1);
  end
j=0;
  for j=(Nf+2:N-1);
  Lnew(1,j)=vap(j-1) + L(1,j-1) + vap(j-2) ;
  end

  for j=1:N-1;
  value1(j) = (abs(L(1,j)-Lnew(1,j)))/L(1,j);
  value2(j)= abs(V(j)-vap(j))/vap(j);
  V(j)=vap(j);
  end
j=0;
  for j=2:N-1
  L(1,j)=Lnew(1,j);
  end
counter2 = counter2 + 1;
convergence = value1

set(handles.text42, 'String', 'Converged!');

end

%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
setappdata(gcbf, 'metricdata', data);
data = getappdata(gcbf, 'metricdata');
data.T_new = T_new;

axes(handles.axes2);
cla;

popup_sel_index = get(handles.popupmenu4, 'Value');
switch popup_sel_index
   case 1
      figure(1);
      plot(T_new,'b:+','LineWidth',2);grid;
      title('Converged Temperature profile'),...
      xlabel('Stage number'),....
      ylabel('Temperature (degC)'),...
      set(handles.text45, 'String', 'Temperature Profile: The trend obtained is similar to two exponential graphs in series connected at the feed stage. The reason for this is the improper placement of the feed stage. In this case, whether the rectifying stage or stripping stage becomes over-staged. ');

     set(handles.text168, 'String','(deg C)');
     set(handles.text145, 'String','');
     set(handles.text146, 'String','');
     set(handles.text147, 'String','');
     set(handles.text148, 'String','');
     set(handles.text149, 'String','');
```

```
set(handles.text150, 'String','');
set(handles.text151, 'String','');
set(handles.text152, 'String','');
set(handles.text153, 'String','');
set(handles.text154, 'String','');
set(handles.text155, 'String','');
set(handles.text156, 'String','');
set(handles.text157, 'String','');
set(handles.text158, 'String','');
set(handles.text159, 'String','');
set(handles.text160, 'String','');

set(handles.text101, 'String',T_new(1));
set(handles.text102, 'String',T_new(2));
set(handles.text103, 'String',T_new(3));
set(handles.text104, 'String',T_new(4));
set(handles.text105, 'String',T_new(5));
set(handles.text106, 'String',T_new(6));
set(handles.text107, 'String',T_new(7));
set(handles.text108, 'String',T_new(8));
set(handles.text109, 'String',T_new(9));
set(handles.text110, 'String',T_new(10));
set(handles.text111, 'String',T_new(11));
set(handles.text112, 'String',T_new(12));
set(handles.text113, 'String',T_new(13));
set(handles.text114, 'String',T_new(14));
set(handles.text115, 'String',T_new(15));
set(handles.text116, 'String',T_new(16));
set(handles.text117, 'String',T_new(17));
set(handles.text118, 'String',T_new(18));
set(handles.text119, 'String',T_new(19));
set(handles.text120, 'String',T_new(20));
set(handles.text121, 'String',T_new(21));
set(handles.text122, 'String',T_new(22));
set(handles.text123, 'String',T_new(23));
set(handles.text124, 'String',T_new(24));
set(handles.text125, 'String',T_new(25));
set(handles.text126, 'String',T_new(26));
set(handles.text127, 'String',T_new(27));
set(handles.text128, 'String',T_new(28));
set(handles.text129, 'String',T_new(29));
set(handles.text130, 'String',T_new(30));
set(handles.text131, 'String',T_new(31));
set(handles.text132, 'String',T_new(32));
set(handles.text133, 'String',T_new(33));
set(handles.text134, 'String',T_new(34));
set(handles.text135, 'String',T_new(35));
set(handles.text136, 'String',T_new(36));
set(handles.text137, 'String',T_new(37));
set(handles.text138, 'String',T_new(38));
set(handles.text139, 'String',T_new(39));
set(handles.text140, 'String',T_new(40));
set(handles.text141, 'String',T_new(41));
set(handles.text142, 'String',T_new(42));
set(handles.text143, 'String',T_new(43));
```

```
    set(handles.text144, 'String',T_new(44));
    set(handles.text145, 'String',T_new(45));
    set(handles.text146, 'String',T_new(46));
    set(handles.text147, 'String',T_new(47));
    set(handles.text148, 'String',T_new(48));
    set(handles.text149, 'String',T_new(49));
    set(handles.text150, 'String',T_new(50));
    set(handles.text151, 'String',T_new(51));
    set(handles.text152, 'String',T_new(52));
    set(handles.text153, 'String',T_new(53));
    set(handles.text154, 'String',T_new(54));
    set(handles.text155, 'String',T_new(55));
    set(handles.text156, 'String',T_new(56));
    set(handles.text157, 'String',T_new(57));
    set(handles.text158, 'String',T_new(58));
    set(handles.text159, 'String',T_new(59));
    set(handles.text160, 'String',T_new(60));
  case 2
    figure(2);
    plot(K_new,'LineWidth',2);grid;
    title('Converged distribution coefficent for each component'),...
    xlabel('Stage number'),....
    ylabel('K value'),...
    LEGEND('propane','isobutane','n-butane','isopentane','n-pentane','n-hexane'),...
    set(handles.text45, 'String', 'From the graph, it is observed that the K value is
proportional to the temperature and pressure of the stage. Also, the K value for the lighter
components, especially propane is a stronger function of temperature and pressure the the K
value for the heavier components. As a result, at the bottom stages, the ratio between the K
values of the lightest and heaviest component is larger than that for the top.');

    set(handles.text168, 'String','(K)');
    set(handles.text145, 'String','');
    set(handles.text146, 'String','');
    set(handles.text147, 'String','');
    set(handles.text148, 'String','');
    set(handles.text149, 'String','');
    set(handles.text150, 'String','');
    set(handles.text151, 'String','');
    set(handles.text152, 'String','');
    set(handles.text153, 'String','');
    set(handles.text154, 'String','');
    set(handles.text155, 'String','');
    set(handles.text156, 'String','');
    set(handles.text157, 'String','');
    set(handles.text158, 'String','');
    set(handles.text159, 'String','');
    set(handles.text160, 'String','');

    set(handles.text101, 'String',ka(1));
    set(handles.text102, 'String',ka(2));
    set(handles.text103, 'String',ka(3));
    set(handles.text104, 'String',ka(4));
    set(handles.text105, 'String',ka(5));
    set(handles.text106, 'String',ka(6));
    set(handles.text107, 'String',ka(7));
```

```
set(handles.text108, 'String',ka(8));
set(handles.text109, 'String',ka(9));
set(handles.text110, 'String',ka(10));
set(handles.text111, 'String',ka(11));
set(handles.text112, 'String',ka(12));
set(handles.text113, 'String',ka(13));
set(handles.text114, 'String',ka(14));
set(handles.text115, 'String',ka(15));
set(handles.text116, 'String',ka(16));
set(handles.text117, 'String',ka(17));
set(handles.text118, 'String',ka(18));
set(handles.text119, 'String',ka(19));
set(handles.text120, 'String',ka(20));
set(handles.text121, 'String',ka(21));
set(handles.text122, 'String',ka(22));
set(handles.text123, 'String',ka(23));
set(handles.text124, 'String',ka(24));
set(handles.text125, 'String',ka(25));
set(handles.text126, 'String',ka(26));
set(handles.text127, 'String',ka(27));
set(handles.text128, 'String',ka(28));
set(handles.text129, 'String',ka(29));
set(handles.text130, 'String',ka(30));
set(handles.text131, 'String',ka(31));
set(handles.text132, 'String',ka(32));
set(handles.text133, 'String',ka(33));
set(handles.text134, 'String',ka(34));
set(handles.text135, 'String',ka(35));
set(handles.text136, 'String',ka(36));
set(handles.text137, 'String',ka(37));
set(handles.text138, 'String',ka(38));
set(handles.text139, 'String',ka(39));
set(handles.text140, 'String',ka(40));
set(handles.text141, 'String',ka(41));
set(handles.text142, 'String',ka(42));
set(handles.text143, 'String',ka(43));
set(handles.text144, 'String',ka(44));
set(handles.text145, 'String',ka(45));
set(handles.text146, 'String',ka(46));
set(handles.text147, 'String',ka(47));
set(handles.text148, 'String',ka(48));
set(handles.text149, 'String',ka(49));
set(handles.text150, 'String',ka(50));
set(handles.text151, 'String',ka(51));
set(handles.text152, 'String',ka(52));
set(handles.text153, 'String',ka(53));
set(handles.text154, 'String',ka(54));
set(handles.text155, 'String',ka(55));
set(handles.text156, 'String',ka(56));
set(handles.text157, 'String',ka(57));
set(handles.text158, 'String',ka(58));
set(handles.text159, 'String',ka(59));
set(handles.text160, 'String',ka(60));
```

```
case 3
    figure(3);
    plot(nn,x1,nn,x2,nn,x3,nn,x4,nn,x5,nn,x6,'LineWidth',2);grid;
    title('Component Distribution in Depropanizer Column'),...
    xlabel('Stage number'),....
    ylabel('Liquid mole fraction'),...
    LEGEND('Propane','Isobutane','N-butane','Iso-pentane','N-pentane','N-hexane'),...
    set(handles.text45, 'String', 'The trend for the propane composition profile (blue line) can
be described as two increasing exponential curves in series that are connected at the feed
stage. However, other components is decreasing as they proceed up the column. The profiles
for the isomers of butane have slight characteristic of being the middle product, as their
composition increases as it proceeds down the column and reduces at the bottom of the
column. ');

    set(handles.text168, 'String','(X)');
    set(handles.text145, 'String','');
    set(handles.text146, 'String','');
    set(handles.text147, 'String','');
    set(handles.text148, 'String','');
    set(handles.text149, 'String','');
    set(handles.text150, 'String','');
    set(handles.text151, 'String','');
    set(handles.text152, 'String','');
    set(handles.text153, 'String','');
    set(handles.text154, 'String','');
    set(handles.text155, 'String','');
    set(handles.text156, 'String','');
    set(handles.text157, 'String','');
    set(handles.text158, 'String','');
    set(handles.text159, 'String','');
    set(handles.text160, 'String','');

    set(handles.text101, 'String',x1(1));
    set(handles.text102, 'String',x1(2));
    set(handles.text103, 'String',x1(3));
    set(handles.text104, 'String',x1(4));
    set(handles.text105, 'String',x1(5));
    set(handles.text106, 'String',x1(6));
    set(handles.text107, 'String',x1(7));
    set(handles.text108, 'String',x1(8));
    set(handles.text109, 'String',x1(9));
    set(handles.text110, 'String',x1(10));
    set(handles.text111, 'String',x1(11));
    set(handles.text112, 'String',x1(12));
    set(handles.text113, 'String',x1(13));
    set(handles.text114, 'String',x1(14));
    set(handles.text115, 'String',x1(15));
    set(handles.text116, 'String',x1(16));
    set(handles.text117, 'String',x1(17));
    set(handles.text118, 'String',x1(18));
    set(handles.text119, 'String',x1(19));
    set(handles.text120, 'String',x1(20));
    set(handles.text121, 'String',x1(21));
    set(handles.text122, 'String',x1(22));
```

```
    set(handles.text123, 'String',x1(23));
    set(handles.text124, 'String',x1(24));
    set(handles.text125, 'String',x1(25));
    set(handles.text126, 'String',x1(26));
    set(handles.text127, 'String',x1(27));
    set(handles.text128, 'String',x1(28));
    set(handles.text129, 'String',x1(29));
    set(handles.text130, 'String',x1(30));
    set(handles.text131, 'String',x1(31));
    set(handles.text132, 'String',x1(32));
    set(handles.text133, 'String',x1(33));
    set(handles.text134, 'String',x1(34));
    set(handles.text135, 'String',x1(35));
    set(handles.text136, 'String',x1(36));
    set(handles.text137, 'String',x1(37));
    set(handles.text138, 'String',x1(38));
    set(handles.text139, 'String',x1(39));
    set(handles.text140, 'String',x1(40));
    set(handles.text141, 'String',x1(41));
    set(handles.text142, 'String',x1(42));
    set(handles.text143, 'String',x1(43));
    set(handles.text144, 'String',x1(44));
    set(handles.text145, 'String',x1(45));
    set(handles.text146, 'String',x1(46));
    set(handles.text147, 'String',x1(47));
    set(handles.text148, 'String',x1(48));
    set(handles.text149, 'String',x1(49));
    set(handles.text150, 'String',x1(50));
    set(handles.text151, 'String',x1(51));
    set(handles.text152, 'String',x1(52));
    set(handles.text153, 'String',x1(53));
    set(handles.text154, 'String',x1(54));
    set(handles.text155, 'String',x1(55));
    set(handles.text156, 'String',x1(56));
    set(handles.text157, 'String',x1(57));
    set(handles.text158, 'String',x1(58));
    set(handles.text159, 'String',x1(59));
    set(handles.text160, 'String',x1(60));


case 4
    figure(4);
    plot(vap,'r:+','LineWidth',2);grid;
    title('Total Vapor Flow Between Stages'),...
    xlabel('Stage number'),....
    ylabel('Flowrate (mol/hour)'),...
    set(handles.text45, 'String', '');

    set(handles.text168, 'String','(mol/hr)');
    set(handles.text145, 'String','');
    set(handles.text146, 'String','');
    set(handles.text147, 'String','');
    set(handles.text148, 'String','');
    set(handles.text149, 'String','');
    set(handles.text150, 'String','');
```

```
set(handles.text151, 'String','');
set(handles.text152, 'String','');
set(handles.text153, 'String','');
set(handles.text154, 'String','');
set(handles.text155, 'String','');
set(handles.text156, 'String','');
set(handles.text157, 'String','');
set(handles.text158, 'String','');
set(handles.text159, 'String','');
set(handles.text160, 'String','');

set(handles.text101, 'String',vap(1));
set(handles.text102, 'String',vap(2));
set(handles.text103, 'String',vap(3));
set(handles.text104, 'String',vap(4));
set(handles.text105, 'String',vap(5));
set(handles.text106, 'String',vap(6));
set(handles.text107, 'String',vap(7));
set(handles.text108, 'String',vap(8));
set(handles.text109, 'String',vap(9));
set(handles.text110, 'String',vap(10));
set(handles.text111, 'String',vap(11));
set(handles.text112, 'String',vap(12));
set(handles.text113, 'String',vap(13));
set(handles.text114, 'String',vap(14));
set(handles.text115, 'String',vap(15));
set(handles.text116, 'String',vap(16));
set(handles.text117, 'String',vap(17));
set(handles.text118, 'String',vap(18));
set(handles.text119, 'String',vap(19));
set(handles.text120, 'String',vap(20));
set(handles.text121, 'String',vap(21));
set(handles.text122, 'String',vap(22));
set(handles.text123, 'String',vap(23));
set(handles.text124, 'String',vap(24));
set(handles.text125, 'String',vap(25));
set(handles.text126, 'String',vap(26));
set(handles.text127, 'String',vap(27));
set(handles.text128, 'String',vap(28));
set(handles.text129, 'String',vap(29));
set(handles.text130, 'String',vap(30));
set(handles.text131, 'String',vap(31));
set(handles.text132, 'String',vap(32));
set(handles.text133, 'String',vap(33));
set(handles.text134, 'String',vap(34));
set(handles.text135, 'String',vap(35));
set(handles.text136, 'String',vap(36));
set(handles.text137, 'String',vap(37));
set(handles.text138, 'String',vap(38));
set(handles.text139, 'String',vap(39));
set(handles.text140, 'String',vap(40));
set(handles.text141, 'String',vap(41));
set(handles.text142, 'String',vap(42));
set(handles.text143, 'String',vap(43));
set(handles.text144, 'String',vap(44));
```

```
set(handles.text145, 'String',vap(45));
set(handles.text146, 'String',vap(46));
set(handles.text147, 'String',vap(47));
set(handles.text148, 'String',vap(48));
set(handles.text149, 'String',vap(49));
set(handles.text150, 'String',vap(50));
set(handles.text151, 'String',vap(51));
set(handles.text152, 'String',vap(52));
set(handles.text153, 'String',vap(53));
set(handles.text154, 'String',vap(54));
set(handles.text155, 'String',vap(55));
set(handles.text156, 'String',vap(56));
set(handles.text157, 'String',vap(57));
set(handles.text158, 'String',vap(58));
set(handles.text159, 'String',vap(59));
set(handles.text160, 'String',vap(60));


case 5
   figure(5);
   plot(L,'b:+','LineWidth',2);grid;
   title('Total Liquid Flow Between Stages'),...
   xlabel('Stage number'),....
   ylabel('Flowrate (mol/hour)'),...
   set(handles.text45, 'String', '');

   set(handles.text168, 'String','(mol/hr)');
   set(handles.text145, 'String','');
   set(handles.text146, 'String','');
   set(handles.text147, 'String','');
   set(handles.text148, 'String','');
   set(handles.text149, 'String','');
   set(handles.text150, 'String','');
   set(handles.text151, 'String','');
   set(handles.text152, 'String','');
   set(handles.text153, 'String','');
   set(handles.text154, 'String','');
   set(handles.text155, 'String','');
   set(handles.text156, 'String','');
   set(handles.text157, 'String','');
   set(handles.text158, 'String','');
   set(handles.text159, 'String','');
   set(handles.text160, 'String','');

   set(handles.text101, 'String',L(1));
   set(handles.text102, 'String',L(2));
   set(handles.text103, 'String',L(3));
   set(handles.text104, 'String',L(4));
   set(handles.text105, 'String',L(5));
   set(handles.text106, 'String',L(6));
   set(handles.text107, 'String',L(7));
   set(handles.text108, 'String',L(8));
   set(handles.text109, 'String',L(9));
   set(handles.text110, 'String',L(10));
   set(handles.text111, 'String',L(11));
```

```
set(handles.text112, 'String',L(12));
set(handles.text113, 'String',L(13));
set(handles.text114, 'String',L(14));
set(handles.text115, 'String',L(15));
set(handles.text116, 'String',L(16));
set(handles.text117, 'String',L(17));
set(handles.text118, 'String',L(18));
set(handles.text119, 'String',L(19));
set(handles.text120, 'String',L(20));
set(handles.text121, 'String',L(21));
set(handles.text122, 'String',L(22));
set(handles.text123, 'String',L(23));
set(handles.text124, 'String',L(24));
set(handles.text125, 'String',L(25));
set(handles.text126, 'String',L(26));
set(handles.text127, 'String',L(27));
set(handles.text128, 'String',L(28));
set(handles.text129, 'String',L(29));
set(handles.text130, 'String',L(30));
set(handles.text131, 'String',L(31));
set(handles.text132, 'String',L(32));
set(handles.text133, 'String',L(33));
set(handles.text134, 'String',L(34));
set(handles.text135, 'String',L(35));
set(handles.text136, 'String',L(36));
set(handles.text137, 'String',L(37));
set(handles.text138, 'String',L(38));
set(handles.text139, 'String',L(39));
set(handles.text140, 'String',L(40));
set(handles.text141, 'String',L(41));
set(handles.text142, 'String',L(42));
set(handles.text143, 'String',L(43));
set(handles.text144, 'String',L(44));
set(handles.text145, 'String',L(45));
set(handles.text146, 'String',L(46));
set(handles.text147, 'String',L(47));
set(handles.text148, 'String',L(48));
set(handles.text149, 'String',L(49));
set(handles.text150, 'String',L(50));
set(handles.text151, 'String',L(51));
set(handles.text152, 'String',L(52));
set(handles.text153, 'String',L(53));
set(handles.text154, 'String',L(54));
set(handles.text155, 'String',L(55));
set(handles.text156, 'String',L(56));
set(handles.text157, 'String',L(57));
set(handles.text158, 'String',L(58));
set(handles.text159, 'String',L(59));
set(handles.text160, 'String',L(60));

case 6
  figure(6);
  plot(T_new,Hl,'b:+',T_new,Hv,'r:o','LineWidth',2);grid;
  title('Liquid and Vapor Molar Enthalpy'),...
  xlabel('Temperature, T (Deg C)'),....
```

```
ylabel('Enthalpy, kJ/mol'),...
LEGEND('Liquid Enthalpy','Vapor Enthalpy')
set(handles.text45, 'String', 'From the plot, it is observed that the molar enthalpy for both ideal gas and liquid mixtures are proportional to temperature. However, the liquid enthalpy is a stronger function of temperature as it has steeper slope.');

set(handles.text168, 'String','');
set(handles.text101, 'String','');
set(handles.text102, 'String','');
set(handles.text103, 'String','');
set(handles.text104, 'String','');
set(handles.text105, 'String','');
set(handles.text106, 'String','');
set(handles.text107, 'String','');
set(handles.text108, 'String','');
set(handles.text109, 'String','');
set(handles.text110, 'String','');
set(handles.text111, 'String','');
set(handles.text112, 'String','');
set(handles.text113, 'String','');
set(handles.text114, 'String','');
set(handles.text115, 'String','');
set(handles.text116, 'String','');
set(handles.text117, 'String','');
set(handles.text118, 'String','');
set(handles.text119, 'String','');
set(handles.text120, 'String','');
set(handles.text121, 'String','');
set(handles.text122, 'String','');
set(handles.text123, 'String','');
set(handles.text124, 'String','');
set(handles.text125, 'String','');
set(handles.text126, 'String','');
set(handles.text127, 'String','');
set(handles.text128, 'String','');
set(handles.text129, 'String','');
set(handles.text130, 'String','');
set(handles.text131, 'String','');
set(handles.text132, 'String','');
set(handles.text133, 'String','');
set(handles.text134, 'String','');
set(handles.text135, 'String','');
set(handles.text136, 'String','');
set(handles.text137, 'String','');
set(handles.text138, 'String','');
set(handles.text139, 'String','');
set(handles.text140, 'String','');
set(handles.text141, 'String','');
set(handles.text142, 'String','');
set(handles.text143, 'String','');
set(handles.text144, 'String','');
set(handles.text145, 'String','');
set(handles.text146, 'String','');
set(handles.text147, 'String','');
set(handles.text148, 'String','');
```

```
set(handles.text149, 'String','');
set(handles.text150, 'String','');
set(handles.text151, 'String','');
set(handles.text152, 'String','');
set(handles.text153, 'String','');
set(handles.text154, 'String','');
set(handles.text155, 'String','');
set(handles.text156, 'String','');
set(handles.text157, 'String','');
set(handles.text158, 'String','');
set(handles.text159, 'String','');
set(handles.text160, 'String','');


case 7
  figure(7);
  plot(nn,Hl,'b:+',nn,Hv,'r:o','LineWidth',2);grid;
  title('Liquid and Vapor Molar Enthalpy '),...
  xlabel('Stage number'),....
  ylabel('Enthalpy, kJ/mol'),...
  LEGEND('Liquid Enthalpy','Vapor Enthalpy')
  set(handles.text45, 'String', 'From the plot, it is observed that the molar enthalpy for both ideal gas and liquid mixtures are proportional to temperature. However, the liquid enthalpy is a stronger function of temperature as it has steeper slope. Refer Tabulated Data for the vapor enthalpy, H (kJ/mol)');

  set(handles.text168, 'String','(kJ/mol)');
  set(handles.text146, 'String','');
  set(handles.text147, 'String','');
  set(handles.text148, 'String','');
  set(handles.text149, 'String','');
  set(handles.text150, 'String','');
  set(handles.text151, 'String','');
  set(handles.text152, 'String','');
  set(handles.text153, 'String','');
  set(handles.text154, 'String','');
  set(handles.text155, 'String','');
  set(handles.text156, 'String','');
  set(handles.text157, 'String','');
  set(handles.text158, 'String','');
  set(handles.text159, 'String','');
  set(handles.text160, 'String','');

  set(handles.text101, 'String',Hv(1));
  set(handles.text102, 'String',Hv(2));
  set(handles.text103, 'String',Hv(3));
  set(handles.text104, 'String',Hv(4));
  set(handles.text105, 'String',Hv(5));
  set(handles.text106, 'String',Hv(6));
  set(handles.text107, 'String',Hv(7));
  set(handles.text108, 'String',Hv(8));
  set(handles.text109, 'String',Hv(9));
  set(handles.text110, 'String',Hv(10));
  set(handles.text111, 'String',Hv(11));
  set(handles.text112, 'String',Hv(12));
```

```
set(handles.text113, 'String',Hv(13));
set(handles.text114, 'String',Hv(14));
set(handles.text115, 'String',Hv(15));
set(handles.text116, 'String',Hv(16));
set(handles.text117, 'String',Hv(17));
set(handles.text118, 'String',Hv(18));
set(handles.text119, 'String',Hv(19));
set(handles.text120, 'String',Hv(20));
set(handles.text121, 'String',Hv(21));
set(handles.text122, 'String',Hv(22));
set(handles.text123, 'String',Hv(23));
set(handles.text124, 'String',Hv(24));
set(handles.text125, 'String',Hv(25));
set(handles.text126, 'String',Hv(26));
set(handles.text127, 'String',Hv(27));
set(handles.text128, 'String',Hv(28));
set(handles.text129, 'String',Hv(29));
set(handles.text130, 'String',Hv(30));
set(handles.text131, 'String',Hv(31));
set(handles.text132, 'String',Hv(32));
set(handles.text133, 'String',Hv(33));
set(handles.text134, 'String',Hv(34));
set(handles.text135, 'String',Hv(35));
set(handles.text136, 'String',Hv(36));
set(handles.text137, 'String',Hv(37));
set(handles.text138, 'String',Hv(38));
set(handles.text139, 'String',Hv(39));
set(handles.text140, 'String',Hv(40));
set(handles.text141, 'String',Hv(41));
set(handles.text142, 'String',Hv(42));
set(handles.text143, 'String',Hv(43));
set(handles.text144, 'String',Hv(44));
set(handles.text145, 'String',Hv(45));
set(handles.text146, 'String',Hv(46));
set(handles.text147, 'String',Hv(47));
set(handles.text148, 'String',Hv(48));
set(handles.text149, 'String',Hv(49));
set(handles.text150, 'String',Hv(50));
set(handles.text151, 'String',Hv(51));
set(handles.text152, 'String',Hv(52));
set(handles.text153, 'String',Hv(53));
set(handles.text154, 'String',Hv(54));
set(handles.text155, 'String',Hv(55));
set(handles.text156, 'String',Hv(56));
set(handles.text157, 'String',Hv(57));
set(handles.text158, 'String',Hv(58));
set(handles.text159, 'String',Hv(59));
set(handles.text160, 'String',Hv(60));


case 8
   figure(8);
   plot(Q,'r:+','LineWidth',2);grid;
   title('Heat Duty in Condenser and Reboiler'),...
   xlabel('Stage number'),....
```

```
ylabel('Heat Flow (kJ/mol)'),...
set(handles.text45, 'String', 'Positive heat flow at stage 1 (reboiler) means the heat in to the system and negative heat flow at stage N (condenser) shows the heat out from the system. It is assume that perfect insulation at the column whereby no heat gain or loss at these intermediate stages.');

set(handles.text168, 'String','(kJ/mol)');
set(handles.text146, 'String','');
set(handles.text147, 'String','');
set(handles.text148, 'String','');
set(handles.text149, 'String','');
set(handles.text150, 'String','');
set(handles.text151, 'String','');
set(handles.text152, 'String','');
set(handles.text153, 'String','');
set(handles.text154, 'String','');
set(handles.text155, 'String','');
set(handles.text156, 'String','');
set(handles.text157, 'String','');
set(handles.text158, 'String','');
set(handles.text159, 'String','');
set(handles.text160, 'String','');

set(handles.text101, 'String',Q(1));
set(handles.text102, 'String',Q(2));
set(handles.text103, 'String',Q(3));
set(handles.text104, 'String',Q(4));
set(handles.text105, 'String',Q(5));
set(handles.text106, 'String',Q(6));
set(handles.text107, 'String',Q(7));
set(handles.text108, 'String',Q(8));
set(handles.text109, 'String',Q(9));
set(handles.text110, 'String',Q(10));
set(handles.text111, 'String',Q(11));
set(handles.text112, 'String',Q(12));
set(handles.text113, 'String',Q(13));
set(handles.text114, 'String',Q(14));
set(handles.text115, 'String',Q(15));
set(handles.text116, 'String',Q(16));
set(handles.text117, 'String',Q(17));
set(handles.text118, 'String',Q(18));
set(handles.text119, 'String',Q(19));
set(handles.text120, 'String',Q(20));
set(handles.text121, 'String',Q(21));
set(handles.text122, 'String',Q(22));
set(handles.text123, 'String',Q(23));
set(handles.text124, 'String',Q(24));
set(handles.text125, 'String',Q(25));
set(handles.text126, 'String',Q(26));
set(handles.text127, 'String',Q(27));
set(handles.text128, 'String',Q(28));
set(handles.text129, 'String',Q(29));
set(handles.text130, 'String',Q(30));
set(handles.text131, 'String',Q(31));
set(handles.text132, 'String',Q(32));
```

```
    set(handles.text133, 'String',Q(33));
    set(handles.text134, 'String',Q(34));
    set(handles.text135, 'String',Q(35));
    set(handles.text136, 'String',Q(36));
    set(handles.text137, 'String',Q(37));
    set(handles.text138, 'String',Q(38));
    set(handles.text139, 'String',Q(39));
    set(handles.text140, 'String',Q(40));
    set(handles.text141, 'String',Q(41));
    set(handles.text142, 'String',Q(42));
    set(handles.text143, 'String',Q(43));
    set(handles.text144, 'String',Q(44));
    set(handles.text145, 'String',Q(45));
    set(handles.text146, 'String',Q(46));
    set(handles.text147, 'String',Q(47));
    set(handles.text148, 'String',Q(48));
    set(handles.text149, 'String',Q(49));
    set(handles.text150, 'String',Q(50));
    set(handles.text151, 'String',Q(51));
    set(handles.text152, 'String',Q(52));
    set(handles.text153, 'String',Q(53));
    set(handles.text154, 'String',Q(54));
    set(handles.text155, 'String',Q(55));
    set(handles.text156, 'String',Q(56));
    set(handles.text157, 'String',Q(57));
    set(handles.text158, 'String',Q(58));
    set(handles.text159, 'String',Q(59));
    set(handles.text160, 'String',Q(60));

  case 9
    figure(9);
    plot(T_new,HLR,'b:+',T_new,HVR,'r:o','LineWidth',2);grid;
    title('Liquid and Vapor Molar Enthalpy(Real Mixture)'),...
    xlabel('Temperature, T (Deg C)'),....
    ylabel('Enthalpy, kJ/mol'),...
    LEGEND('Liquid Enthalpy','Vapor Enthalpy')
    set(handles.text45, 'String', 'From the plot, it is observed that the molar enthalpy for both
ideal gas and liquid mixtures are proportional to temperature. However, the liquid enthalpy is
a stronger function of temperature as it has steeper slope.');

    set(handles.text168, 'String','');
    set(handles.text101, 'String','');
    set(handles.text102, 'String','');
    set(handles.text103, 'String','');
    set(handles.text104, 'String','');
    set(handles.text105, 'String','');
    set(handles.text106, 'String','');
    set(handles.text107, 'String','');
    set(handles.text108, 'String','');
    set(handles.text109, 'String','');
    set(handles.text110, 'String','');
    set(handles.text111, 'String','');
    set(handles.text112, 'String','');
    set(handles.text113, 'String','');
    set(handles.text114, 'String','');
```

```
    set(handles.text115, 'String','');
    set(handles.text116, 'String','');
    set(handles.text117, 'String','');
    set(handles.text118, 'String','');
    set(handles.text119, 'String','');
    set(handles.text120, 'String','');
    set(handles.text121, 'String','');
    set(handles.text122, 'String','');
    set(handles.text123, 'String','');
    set(handles.text124, 'String','');
    set(handles.text125, 'String','');
    set(handles.text126, 'String','');
    set(handles.text127, 'String','');
    set(handles.text128, 'String','');
    set(handles.text129, 'String','');
    set(handles.text130, 'String','');
    set(handles.text131, 'String','');
    set(handles.text132, 'String','');
    set(handles.text133, 'String','');
    set(handles.text134, 'String','');
    set(handles.text135, 'String','');
    set(handles.text136, 'String','');
    set(handles.text137, 'String','');
    set(handles.text138, 'String','');
    set(handles.text139, 'String','');
    set(handles.text140, 'String','');
    set(handles.text141, 'String','');
    set(handles.text142, 'String','');
    set(handles.text143, 'String','');
    set(handles.text144, 'String','');
    set(handles.text145, 'String','');
    set(handles.text146, 'String','');
    set(handles.text147, 'String','');
    set(handles.text148, 'String','');
    set(handles.text149, 'String','');
    set(handles.text150, 'String','');
    set(handles.text151, 'String','');
    set(handles.text152, 'String','');
    set(handles.text153, 'String','');
    set(handles.text154, 'String','');
    set(handles.text155, 'String','');
    set(handles.text156, 'String','');
    set(handles.text157, 'String','');
    set(handles.text158, 'String','');
    set(handles.text159, 'String','');
    set(handles.text160, 'String','');

case 10
    figure(10);
    plot(nn,HLR,'b:+',nn,HVR,'r:o','LineWidth',2);grid;
    title('Liquid and Vapor Molar Enthalpy '),...
    xlabel('Stage number'),....
    ylabel('Enthalpy, kJ/mol'),...
    LEGEND('Liquid Enthalpy','Vapor Enthalpy')
```

```
set(handles.text45, 'String', 'From the plot, it is observed that the molar enthalpy for both ideal gas and liquid mixtures are proportional to temperature. However, the liquid enthalpy is a stronger function of temperature as it has steeper slope. Refer Tabulated Data for the vapor enthalpy, H (kJ/mol)');

set(handles.text168, 'String','(kJ/mol)');
set(handles.text146, 'String','');
set(handles.text147, 'String','');
set(handles.text148, 'String','');
set(handles.text149, 'String','');
set(handles.text150, 'String','');
set(handles.text151, 'String','');
set(handles.text152, 'String','');
set(handles.text153, 'String','');
set(handles.text154, 'String','');
set(handles.text155, 'String','');
set(handles.text156, 'String','');
set(handles.text157, 'String','');
set(handles.text158, 'String','');
set(handles.text159, 'String','');
set(handles.text160, 'String','');

set(handles.text101, 'String',HVR(1));
set(handles.text102, 'String',HVR(2));
set(handles.text103, 'String',HVR(3));
set(handles.text104, 'String',HVR(4));
set(handles.text105, 'String',HVR(5));
set(handles.text106, 'String',HVR(6));
set(handles.text107, 'String',HVR(7));
set(handles.text108, 'String',HVR(8));
set(handles.text109, 'String',HVR(9));
set(handles.text110, 'String',HVR(10));
set(handles.text111, 'String',HVR(11));
set(handles.text112, 'String',HVR(12));
set(handles.text113, 'String',HVR(13));
set(handles.text114, 'String',HVR(14));
set(handles.text115, 'String',HVR(15));
set(handles.text116, 'String',HVR(16));
set(handles.text117, 'String',HVR(17));
set(handles.text118, 'String',HVR(18));
set(handles.text119, 'String',HVR(19));
set(handles.text120, 'String',HVR(20));
set(handles.text121, 'String',HVR(21));
set(handles.text122, 'String',HVR(22));
set(handles.text123, 'String',HVR(23));
set(handles.text124, 'String',HVR(24));
set(handles.text125, 'String',HVR(25));
set(handles.text126, 'String',HVR(26));
set(handles.text127, 'String',HVR(27));
set(handles.text128, 'String',HVR(28));
set(handles.text129, 'String',HVR(29));
set(handles.text130, 'String',HVR(30));
set(handles.text131, 'String',HVR(31));
set(handles.text132, 'String',HVR(32));
set(handles.text133, 'String',HVR(33));
```

```
    set(handles.text134, 'String',HVR(34));
    set(handles.text135, 'String',HVR(35));
    set(handles.text136, 'String',HVR(36));
    set(handles.text137, 'String',HVR(37));
    set(handles.text138, 'String',HVR(38));
    set(handles.text139, 'String',HVR(39));
    set(handles.text140, 'String',HVR(40));
    set(handles.text141, 'String',HVR(41));
    set(handles.text142, 'String',HVR(42));
    set(handles.text143, 'String',HVR(43));
    set(handles.text144, 'String',HVR(44));
    set(handles.text145, 'String',HVR(45));
    set(handles.text146, 'String',HVR(46));
    set(handles.text147, 'String',HVR(47));
    set(handles.text148, 'String',HVR(48));
    set(handles.text149, 'String',HVR(49));
    set(handles.text150, 'String',HVR(50));
    set(handles.text151, 'String',HVR(51));
    set(handles.text152, 'String',HVR(52));
    set(handles.text153, 'String',HVR(53));
    set(handles.text154, 'String',HVR(54));
    set(handles.text155, 'String',HVR(55));
    set(handles.text156, 'String',HVR(56));
    set(handles.text157, 'String',HVR(57));
    set(handles.text158, 'String',HVR(58));
    set(handles.text159, 'String',HVR(59));
    set(handles.text160, 'String',HVR(60));


  case 11
    figure(11);
    plot(QRR,'r:+','LineWidth',2);grid;
    title('Heat Duty in Condenser and Reboiler'),...
    xlabel('Stage number'),....
    ylabel('Heat Flow (kJ/mol)'),...
    set(handles.text45, 'String', 'Positive heat flow at stage 1 (reboiler) means the heat in to
the system and negative heat flow at stage N (condenser) shows the heat out from the system.
It is assume that perfect insulation at the column whereby no heat gain or loss at these
intermediate stages.');

    set(handles.text168, 'String','(kJ/mol)');
    set(handles.text146, 'String','');
    set(handles.text147, 'String','');
    set(handles.text148, 'String','');
    set(handles.text149, 'String','');
    set(handles.text150, 'String','');
    set(handles.text151, 'String','');
    set(handles.text152, 'String','');
    set(handles.text153, 'String','');
    set(handles.text154, 'String','');
    set(handles.text155, 'String','');
    set(handles.text156, 'String','');
    set(handles.text157, 'String','');
    set(handles.text158, 'String','');
    set(handles.text159, 'String','');
```

```
set(handles.text160, 'String','');

set(handles.text101, 'String',QRR(1));
set(handles.text102, 'String',QRR(2));
set(handles.text103, 'String',QRR(3));
set(handles.text104, 'String',QRR(4));
set(handles.text105, 'String',QRR(5));
set(handles.text106, 'String',QRR(6));
set(handles.text107, 'String',QRR(7));
set(handles.text108, 'String',QRR(8));
set(handles.text109, 'String',QRR(9));
set(handles.text110, 'String',QRR(10));
set(handles.text111, 'String',QRR(11));
set(handles.text112, 'String',QRR(12));
set(handles.text113, 'String',QRR(13));
set(handles.text114, 'String',QRR(14));
set(handles.text115, 'String',QRR(15));
set(handles.text116, 'String',QRR(16));
set(handles.text117, 'String',QRR(17));
set(handles.text118, 'String',QRR(18));
set(handles.text119, 'String',QRR(19));
set(handles.text120, 'String',QRR(20));
set(handles.text121, 'String',QRR(21));
set(handles.text122, 'String',QRR(22));
set(handles.text123, 'String',QRR(23));
set(handles.text124, 'String',QRR(24));
set(handles.text125, 'String',QRR(25));
set(handles.text126, 'String',QRR(26));
set(handles.text127, 'String',QRR(27));
set(handles.text128, 'String',QRR(28));
set(handles.text129, 'String',QRR(29));
set(handles.text130, 'String',QRR(30));
set(handles.text131, 'String',QRR(31));
set(handles.text132, 'String',QRR(32));
set(handles.text133, 'String',QRR(33));
set(handles.text134, 'String',QRR(34));
set(handles.text135, 'String',QRR(35));
set(handles.text136, 'String',QRR(36));
set(handles.text137, 'String',QRR(37));
set(handles.text138, 'String',QRR(38));
set(handles.text139, 'String',QRR(39));
set(handles.text140, 'String',QRR(40));
set(handles.text141, 'String',QRR(41));
set(handles.text142, 'String',QRR(42));
set(handles.text143, 'String',QRR(43));
set(handles.text144, 'String',QRR(44));
set(handles.text145, 'String',QRR(45));
set(handles.text146, 'String',QRR(46));
set(handles.text147, 'String',QRR(47));
set(handles.text148, 'String',QRR(48));
set(handles.text149, 'String',QRR(49));
set(handles.text150, 'String',QRR(50));
set(handles.text151, 'String',QRR(51));
set(handles.text152, 'String',QRR(52));
set(handles.text153, 'String',QRR(53));
```

```
    set(handles.text154, 'String',QRR(54));
    set(handles.text155, 'String',QRR(55));
    set(handles.text156, 'String',QRR(56));
    set(handles.text157, 'String',QRR(57));
    set(handles.text158, 'String',QRR(58));
    set(handles.text159, 'String',QRR(59));
    set(handles.text160, 'String',QRR(60));


  case 12
    figure(12);
    surf(peaks);
    set(handles.text45, 'String', '');

end

function popupmenu2_CreateFcn(hObject, eventdata, handles)
if ispc
  set(hObject,'BackgroundColor','white');
else
  set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
set(hObject, 'String', {'Temperature', 'Equilibrium Constant, K', 'Component
Distribution','Vapor Flow', 'Liquid Flow','Enthalpy vs Temperature','Liquid and Vapor
Enthalpy','Heat Duty','Demo'});

function popupmenu2_Callback(hObject, eventdata, handles)
function pushbutton26_Callback(hObject, eventdata, handles)
function pushbutton27_Callback(hObject, eventdata, handles)
function pushbutton28_Callback(hObject, eventdata, handles)
function pushbutton29_Callback(hObject, eventdata, handles)
function pushbutton31_Callback(hObject, eventdata, handles)
function pushbutton32_Callback(hObject, eventdata, handles)
function pushbutton33_Callback(hObject, eventdata, handles)
function pushbutton34_Callback(hObject, eventdata, handles)
function pushbutton35_Callback(hObject, eventdata, handles)
function pushbutton36_Callback(hObject, eventdata, handles)
function pushbutton37_Callback(hObject, eventdata, handles)
function pushbutton38_Callback(hObject, eventdata, handles)
function pushbutton39_Callback(hObject, eventdata, handles)
function pushbutton40_Callback(hObject, eventdata, handles)
function pushbutton45_Callback(hObject, eventdata, handles)
function pushbutton46_Callback(hObject, eventdata, handles)
function pushbutton47_Callback(hObject, eventdata, handles)
function pushbutton48_Callback(hObject, eventdata, handles)
function pushbutton49_Callback(hObject, eventdata, handles)
function pushbutton50_Callback(hObject, eventdata, handles)
function pushbutton51_Callback(hObject, eventdata, handles)
function pushbutton52_Callback(hObject, eventdata, handles)
function pushbutton59_Callback(hObject, eventdata, handles)
function pushbutton60_Callback(hObject, eventdata, handles)
function pushbutton61_Callback(hObject, eventdata, handles)
function pushbutton62_Callback(hObject, eventdata, handles)
function pushbutton63_Callback(hObject, eventdata, handles)
function pushbutton64_Callback(hObject, eventdata, handles)
```

```
set(handles.text45, 'String', 'Total Condenser : All of the vapor leaving the top of the column
is condensed. Consequently, the composition of the vapor leaving the top tray y(N-1) is the
same as that of the liquid distillate product and reflux, xD.');
function pushbutton65_Callback(hObject, eventdata, handles)
set(handles.text45, 'String', 'Partial Reboiler : It vaporize part of the liquid in the column base.
Partial reboilers also provide an ideal separation stage. Reboilers may be "thermosiphon"
types that rely on the thermal effects on density to draw liquid through the heat exchanger,
"forced circulation" types that use a pump to force liquid through, or even "stab-in" types that
come through the side of the column into the liquid reservoir.');
function pushbutton66_Callback(hObject, eventdata, handles)
function pushbutton67_Callback(hObject, eventdata, handles)
function pushbutton68_Callback(hObject, eventdata, handles)
function pushbutton69_Callback(hObject, eventdata, handles)
function pushbutton70_Callback(hObject, eventdata, handles)
function pushbutton71_Callback(hObject, eventdata, handles)
function pushbutton76_Callback(hObject, eventdata, handles)
function pushbutton77_Callback(hObject, eventdata, handles)
function pushbutton78_Callback(hObject, eventdata, handles)
function pushbutton79_Callback(hObject, eventdata, handles)
function edit30_CreateFcn(hObject, eventdata, handles)

if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function edit30_Callback(hObject, eventdata, handles)
function pushbutton110_Callback(hObject, eventdata, handles)
function pushbutton111_Callback(hObject, eventdata, handles)

%@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

function popupmenu4_CreateFcn(hObject, eventdata, handles)

if ispc
   set(hObject,'BackgroundColor','white');
else
   set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

function popupmenu4_Callback(hObject, eventdata, handles)


% --- Executes on button press in pushbutton114.
function pushbutton114_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton114 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton113.
function pushbutton113_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to pushbutton113 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton117.
function pushbutton117_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton117 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)


% --- Executes on button press in pushbutton118.
function pushbutton118_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton118 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```