

Modeling and Simulation of an Isomerization Reactor using OpenModelica

Submitted in partial fulfillment of the requirements
of the degree of

Master of Technology

by

Venkata Sai Pavan Chetan Annam
(Roll No. 203020008)

Supervisors:

Prof. Kannan Moudgalya
Prof. Sanjay M. Mahajani



Department of Chemical Engineering
INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
2022

Thesis Approval

This thesis entitled **Modeling and Simulation of an Isomerization Reactor using OpenModelica** by **Venkata Sai Pavan Chetan Annam** is approved for the degree of **Master of Technology**.

Examiners:

.....
.....
.....
.....
.....

Supervisor:

Chairperson:

Date: June 20, 2022

Place: IIT Bombay

Declaration

I declare that this written submission represents my ideas in my own words and where others ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date:June 20, 2022

Venkata Sai Pavan Chetan Annam

Roll No. 203020008

Acknowledgments

I would like to express my sincere gratitude to my guide, **Prof. Kannan Moudgalya** for his continuous support by pushing my limits. I also like to express my thankfulness to my guide, **Prof. Sanjay M. Mahajani** for his help during the weekly meetings. The academics taught at IIT Bombay has helped me a lot in doing this project and I'm always indebted to the Chemical Engineering Department for providing such an opportunity. Thanks to Priyam Naik (PHD student) for helping me in finding the thermodynamics part easily in the DWSIM and also in scrutinizing the report. Nikhil Sharma and Praveen Dalve are always there when I face an issue with the Open Modelica. Last but not least, I really thank Digvijay from FOSSEE for helping me in the interoperability part.

Abstract

In this work Isomerization reactor is modeled as a Plug Flow Reactor (PFR) involving 17 components, comprising C6 and lower molecular weight hydrocarbons and Hydrogen, participating through 54 reactions. This system is modelled with 18 Ordinary Differential Equations. This is an ODE-IVP problem and is solved using an OpenModelica solver called DASSL. We have found that OpenModelica computes more than 100 times faster than Aspen-Hysys's PFR model. Model predictions are made to match literature profiles by tuning the activation energies of 54 reactions. Both vapor phase and liquid phase reactions are considered.

Function based optimization is used to automatically tune the model parameters. The optimization procedure is automated by integrating the simulation model with an optimizer COBYLA, readily available in the R statistical software. The ability to quickly integrate model equations by DASSL in OpenModelica allows a large number of function evaluations in a short time, enabling function based optimization. Mean Square Error criterion has been found to be better than Absolute Error criterion to tune the model parameters. After tuning, model predictions are in agreement with the literature findings. No detailed effort has been taken to reduce the number of tuning parameters, however.

Contents

Abstract	i
List of Tables	vii
List of Figures	ix
List of Abbreviations	xv
List of Symbols	xvii
1 Introduction	1
2 Isomerization Unit Overview	5
2.1 Feed Characterization	6
2.2 Chemistry	7
2.2.1 Isomerization Reaction	7
2.2.2 Benzene Saturation	8
2.2.3 Ring Opening of Naphthenes	8
2.2.4 Hydro Cracking Reactions	8
2.3 Thermodynamics	9
2.4 Catalysis	9

2.5	Operating Conditions	10
2.6	Performance Numbers	11
2.6.1	Product Iso Ratios	12
2.6.2	Paraffin Isomerization	13
2.6.3	Octane Number	13
2.7	Isomerization Processes	14
2.7.1	Once Through Process	14
2.7.2	Deisohexanizer process	14
2.7.3	Deisopentanizer with Deisohexanizer process	15
3	Vapor Phase Modeling of the Reactor	17
3.1	Component Balance	17
3.2	The Importance of Rate Constant (k_j)	18
3.3	Energy Balance	20
3.4	Thermodynamic Modeling	21
3.5	OpenModelica	22
3.6	Simulation Results	24
4	Function based Optimizer in OpenModelica	31
4.1	COBYLA Optimizer	33
4.2	Providing COBYLA Optimizer in OpenModelica through Interoperability	34

4.2.1	Some Basic Commands of OpenModelica	34
4.2.2	Interoperability of R Software and OpenModelica	36
5	Optimization of vapor phase model	39
5.1	Absolute Mean Error	40
5.2	Mean Square Error (MSE)	42
5.3	Reducing the Number of Parameters	45
6	Liquid Phase Reactor Simulation	51
6.1	Optimization of Liquid Phase Model	52
7	Conclusion and Future work	59
References		61
Appendix A	Source Codes	63
A.1	OpenModelica Code	63
A.2	R-Language Code	72
A.3	Reactions	74
Appendix B	Interoperability of DWSIM and Open Modelica	75
B.1	Concept	75
B.2	Source Code	76

List of Tables

2.1	Typical inlet and outlet composition of an ISOM reactor (Garg, class notes)	6
2.2	Summary of Catalysts used in an Isomerization reactor (Valavarasu and Sairam, 2013)	11
2.3	Typical operating conditions based of an ISOM reactor based on catalyst (Valavarasu and Sairam, 2013)	12
3.1	Table showing the Design parameters used in the vapor phase model	24
3.2	Table showing the operating conditions of material stream used in the Vapor phase model(Enikeeva et al., 2021)	24
3.3	Table showing the initial kinetic parameters for the vapor phase model (Enikeeva et al., 2021)	26
3.4	Table showing the list of components and their composition used in the vapor phase model (Enikeeva et al., 2021)	27
5.1	Table showing the performance parameters for Absolute Error as objective function.	43
5.2	Table showing the performance parameters for MSE as objective function.	46
5.3	Table showing the converged values of activation energies and their deviation from the original parameters given in Table 3.3 on Page 26 (Enikeeva et al., 2021). MSE is used for tuning.	49
6.1	Table showing the design parameters of the ISOM reactor (Aspen-Hysys)	52

6.2	Table showing the operating conditions of the stream entering the reactor (Aspen-Hysys)	52
6.3	Table showing the composition of the stream entering the ISOM reactor (Aspen-Hysys)	53
6.4	Table showing the comparison of performance parameters between Aspen-Hysys Simulation and OM Model after tuning.	58

List of Figures

2.1	Effect of yield on C6 components upon increasing the temperature (Garg-class notes).	10
2.2	Effect of temperature with and without recycle on isomerate octane number (Garg-class notes).	12
2.3	Flowsheet showing the Once Through process (Garg-class notes).	15
2.4	Isomerization unit containing the Deisohexanizer (Garg-class notes).	15
2.5	Isomerization unit installed with the Deisopentanizer and hte Deisohexanizer (Garg-class notes).	16
3.1	Reaction Model used, these reactions can be divided into 4 types of reactions as discussed in section 2.2 (Enikeeva et al., 2021)	18
3.2	Figure showing the breakdown of how heat of reaction is calculated at elevated temperatures(Hess Law) (Smith et al., 2005)	20
3.3	Figure showing the schema of reactors arranged in Enikeeva et al. (2021)	25
3.4	Figure showing the plots of n-parffins of across the reactor length. The solid lines represent the simulation results and the points represent the Enikeeva et al. (2021) data. Vertical grid lines showing the reactor end.	28
3.5	Figure showing the plots of value added components across the reactor length. The solid lines represent the simulation results and the points represent the Enikeeva et al. (2021) data. Vertical grid lines showing the reactor end.	28

3.6	Figure showing the plots of 2MP and 3MP weight fraction across the reactor length. The deviation is due to the different model considered here. The solid lines represent the simulation results and the points represent the Enikeeva et al. (2021) data. Vertical grid lines showing the reactor end.	29
3.7	Figure showing the comparison plots of temperature profile across the reactor. The results are not matching with the Enikeeva et al. (2021) data because we have different model considered in this work. The sudden drop in the temperature profile at 1.46m is due to the cooler. The solid lines represent the simulation results and the points represent the Enikeeva et al. (2021) data.	29
3.8	Bar graph showing the comparison of outlet weight fraction between literature and OM model.	30
4.1	Typical algorithm to solve an optimization problem	32
4.2	Algorithm of the COBYLA optimizer (Powell, 1994)	35
4.3	Figure showing the interoperability of R-lang and OpenModelica	37
5.1	Figure showing the concentration profile of n-paraffins across the reactor length after tuning with Absolute Error as objective function. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.	41
5.2	Figure showing the concentration profile of value added components across the reactor length after tuning with Absolute Error as objective function. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.	41
5.3	Figure showing the concentration profile of 2MP and 3MP across the reactor length after tuning with Absolute Error as objective function. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.	42

5.4	Figure showing temperature profile across the reactor length after tuning with Absolute Error as objective function. The sudden drop in the temperature profile at 1.46m is due to the cooler. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.	42
5.5	Figure showing the comparison of outlet composition(weight fraction) from both Enikeeva et al. (2021) and Model after tuning with Absolute Error as objective function. The optimization objective function contains the end point data only.	43
5.6	Figure showing the concentration profile of n-paraffins across the reactor length for the end point tuning. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.	44
5.7	Figure showing the concentration profile of value added components across the reactor length after tuning with Mean Square Error as objective function. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.	44
5.8	Figure showing the concentration profile of 2MP and 3MP across the reactor length after tuning with Mean Square Error as objective function. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.	45
5.9	Figure showing temperature profile across the reactor length after tuning with Mean Square Error as objective function. The sudden drop in the temperature profile at 1.46m is due to the cooler. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.	45

5.10	Figure showing the comparison of outlet composition(weight fraction) from both Enikeeva et al. (2021) and Model after tuning with Mean Square Error as objective function. The optimization objective function contains the end point data only.	46
5.11	Figure showing the concentration profile of n-paraffins across the reactor length after tuning with selected parameters. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.	47
5.12	Figure showing the concentration profile of value added components across the reactor length after tuning with selected parameters. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.	48
5.13	Figure showing the concentration profile of 2MP and 3MP across the reactor length after tuning with selected parameters.The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.	48
5.14	Figure showing temperature profile across the reactor length after tuning with selected parameters. The sudden drop in the temperature profile at 1.46m is due to the cooler. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.	50
5.15	Figure showing the comparison of outlet composition(weight fraction) from both Enikeeva et al. (2021) and Model after tuning with selected parameters.	50
6.1	comparison plots of n-paraffins between Aspen-Hysys and OM model. Here, vertical bar represents the end of the reactor. Solid lines represent OM results and the dots represents Aspen-Hysys data.	54

6.2	Comparison plots of value added components obtained between Aspen-Hysys and OM model. Here, vertical bar represents the end of the reactor. Solid lines represent OM results and the dots represents Aspen-Hysys data.	54
6.3	Comparison plots of 2MP and 3Mp between Aspen-Hysys and OM model. Here, vertical bar represents the end of the reactor. Solid lines represent OM results and the dots represents Aspen-Hysys data.	55
6.4	Comparison plots of Temperature profile between Aspen-Hysys and OM model. Here, vertical bar represents the end of the reactor. Solid lines represent OM results and the dots represents Aspen-Hysys data.	55
6.5	Figure showing the concentration profile of n-paraffins across the reactor length after tuning. The solid lines represent the simulation results and the points represent the data from Aspen-Hysys. Vertical grid lines showing the reactor end.	56
6.6	Figure showing the concentration profile of value added components across the reactor length after tuning. The solid lines represent the simulation results and the points represent the data from the Aspen-Hysys Simulation. Vertical grid lines showing the reactor end.	56
6.7	Figure showing the concentration profile of 2MP and 3MP across the reactor length after tuning. The solid lines represent the simulation results and the points represent the data from the Aspen-Hysys Simulation. Vertical grid lines showing the reactor end.	57
6.8	Figure showing temperature profile across the reactor length after tuning. The sudden drop in the temperature profile at 1.46m is due to the cooler. The solid lines represent the simulation results and the points represent the data from Aspen-Hysys Simulation. Vertical grid lines showing the reactor end.	57
6.9	Figure showing the comparison of outlet composition(weight fraction) from both Aspen-Hysys Simulation and OM Model after tuning.	58

A.1	The reactions considered in this work (Enikeeva et al., 2021)	74
B.1	Figure showing the usage of custom Modeling in DWSIM	76

List of Abbreviations

C4-	hydrocarbons with carbon atoms less than or equal to 4
C5	represents hydrocarbons containing 5 carbon atoms
C6	repesents hydrocarbons containing 6 carbons atoms
nC5	n-pentane
nC6	n-hexane
iC5	isopentane
C5P	represents C5 paraffins
C6p	represents C6 paraffins
2MP	2methylpentane
3MP	3methylpentane
2,2DMB	2,2dimethyl butane
2,3DMB	2,3dimethyl butane
CH	cylcohexane
H₂	Hydrogen
MCP	methyl cyclopentane
C7+	hydrocarbons with carbon atoms more than or equal to 7
LHSV	liquid hourly space velocity
RON	research octane number
RONi	research octane number of component i
OM	Open Modelica
ODE	ordinary differential equation's
ACS	area of cross section (m^2)
PIN	paraffin isomerization number

DAE	differential algebraic equation's
DASSL	differential algebraic system solver
PFR	plug flow reactor

List of Symbols

y_i	mole fraction of component i
C_i	concentration of component i (mol/m^3)
t	residence time (hr)
q	volumetric flow rate (m^3/hr)
L	length of the reactor (m)
r_j	rate of reaction j ($kmol/m^3 - hr$)
k_j	rate constant of reaction j
k_{j0}	pre-exponential factor of reaction j (hr^{-1})
E_j	activation energy of reaction j ($kJ/kmol$)
η	effectivness factor
H	Henry's constant ($kPa - m^3/kmol$)
T	temperature in the reactor (K)
Q_j	heat of reaction at temperature T ($kJ/kmol$)
Ca_m	concentration of the reaction mixture (mol/m^3)
C_{p_m}	heat capacity of the reaction mixture ($kJ/kmol - {}^\circ K$)
ΔH_R	amount of heat required to cool the reactants from temperature T to T_0
ΔH_P	amount of heat requiried to heat the reactants from the temperature T_0 to T
Z	compressibility factor
P	pressure (kPa)
$C_{p_m}^{res}$	residual heat capacity of reaction mixture ($kJ/kmol - {}^\circ K$)
$C_{p_m}^{id}$	ideal heat capacity of reaction mixture ($kJ/kmol - {}^\circ K$)
V	molar volume (m^3/mol)
\bar{x}	a vector of parameters that need to be optimized

\bar{x}^{new}	updated vector of parameters
θ	reflection constant
$F(\bar{x})$	objective function which is a function of parameter vector \bar{x}
$C_i(\bar{x})$	Constraint i
ϕ	error function in COBYLA optimizer
ρ_{beg}	initial rho value given by the user which is used to generate simplex.
ρ	trust region.
ρ_{end}	final trust region value given by user.

Chapter 1

Introduction

There are many side cuts or streams obtained from the Crude Distillation unit which need to be processed or refined before end-use. One such stream is Naphtha. Naphtha is dealt with in the “Naphtha Complex” a section in a refinery. Naphtha is split into two streams using a naphtha splitter, namely Light Naphtha and Heavy naphtha. These streams are treated separately in units namely, the Isomerization unit and Catalytic reformer unit. The main objective of the Naphtha complex refinery is to improve the octane number. The boiling range of C5/C6 Light Naphtha is 27° to 70° C. This stream has an Octane Number of 70, which can be improved by increasing branched-chain paraffin. Branched chains have a higher Octane number when compared to their corresponding straight-chain paraffin. The Isomerization unit improves the octane number of Light Naphtha by harnessing the ability of the isomerization chemical reaction to produce branched-chain paraffin from straight-chain paraffin. Heavy Naphtha is processed in a Catalytic reformer. Dehydrogenation reaction is favored in this unit, hence the formation of aromatics and olefins is desired. These compounds also have higher octane numbers, so this unit also helps in increasing the octane number.

The output obtained from the ISOM unit called Isomerate is blended with another stream called Reformate from the Catalytic Reformer to form a gasoline pool. Benzene is a carci-

genic compound and is not desired in gasoline. Hence the norms imposed by Bharat-Stage. ISOM unit helps in regulating the benzene content in the gasoline pool by saturating it. And all the benzene precursors are directed into the Light Naphtha stream in the Naphtha splitter without letting them into the Heavy naphtha stream. Benzene precursors are those components that form Benzene easily in the dehydrogenating environment. Because, if any of the benzene precursors lands in the heavy naphtha, the Catalytic reformer converts them into Benzene and it'll end up in the final gasoline pool which is not desired.

The modeling of the reaction network for the Isomerization reactor was started by Chekantsev et al. (2014) called universal modeling. This work adapted the reaction model from Enikeeva et al. (2021). C7+ components and their reactions are not considered in this work. The reactor model assumed is the Plug flow model. Peng Robinson thermodynamic calculations are used for calculating compressibility factors, heat capacity, etc. Catalyst activity is not modeled because they get buried under the rate constant, these things are discussed in detail in the later sections. Tuning of kinetic parameters is done using an optimizer named COBYLA.

Public sector units in India had approached the center of Excellence in Oil Gas and Energy, IIT Bombay to develop a refinery package in open source software for some reasons: 1) Commercial packages are expensive hence they add up to the operating costs, leading to less profit. 2) Government approvals are required to get more licenses, and it takes a lot of time. Because of these reasons, public sector companies wanted to develop an open-source software package for refinery applications.

Open source simulators like DWSIM and OpenModelica these days have developed a lot and can compete with commercial packages like Aspen and Dymola. The importance of building this model is that almost 50 reactions happen inside the ISOM reactor and one can't expect a user to add 50 reactions manually to the simulator. Tuning parameters manually is almost impossible. Tuning of parameters is useful as they help in facing a new situation.

We used OpenModelica and R programming languages in this work to build the ISOM Reactor. Aspen-Hysys has its package called Isomerization reactor, which can process the reactions and tune the kinetic parameters. This project also aims at building the same kind of packages with open-source software. Some of the objectives which we try to achieve in this work are:

- To create an OpenModelica package for the ISOM reactor using OpenModelica.
- Since the manual tuning of parameters is tedious, an optimizer has to be used to tune the kinetic parameters of an ISOM reactor. OpenModelica does not have an optimization facility and hence the COBYLA package available in R-language is used. So, interoperability of R-language and OpenModelica should be achieved.
- Initially, the model will be tried upon the data from literature and tuned the parameters.

Chapter-2 is an epitome of an ISOM unit. It explains feed characterization, the chemistry involved in the reactors, operating conditions, catalysis, and various flowsheet schema used in the ISOM unit. Chapter-3 gives the detailed modeling of the vapor phase reactor model and the simulation results produced using OpenModelica. It also contains the briefing on OpenModelica software and why it is helpful in this work. Chapter-4 indicates the importance of an function based evaluation optimizer and explains about the COBYLA optimizer. Integration of the OpenModelica software with the R software is also explained in this chapter. Because COBYLA package is readily available with the R software. Chapter-5 shows the results obtained after tuning the parameters and also explains the importance of selecting a good objective function. Chapter-6 contains the Liquid phase reactor simulation. Conclusion and Future work are presented in Chapter-7.

This page was intentionally left blank.

Chapter 2

Isomerization Unit Overview

Octane number decides the quality of gasoline. The higher the number, the higher is the performance of an Internal Combustion Engine. To increase the octane number of gasoline, a light quantity of tetraethyl lead (TEL) is mixed in the gasoline pool. It is called an anti-knocking agent because it improves performance by increasing the octane number. As the side effects caused by tetra ethyl lead have come into the light, countries started to put a ban. Also, limitations are imposed upon the usage of aromatics and olefins in gasoline. Since olefins and aromatics have higher octane numbers, their restriction leads to a decrease in the Octane number of the gasoline. Earlier ISOM unit is only used for the production of isobutane using acidic catalysts, and now it is used to improve the Octane number of Light Naphtha stream by developing dual functional catalyst (Valavarasu and Sairam, 2013). We ultimately use ISOM unit to improve the Octane number of Light Naphtha.

This chapter explains the feed characterization, chemistry, catalysis, and thermodynamics of the Isomerization unit. Various schema has evolved with time to improve the yield and the octane number of Isomerate (outlet of the reactor) is also discussed.

2.1 Feed Characterization

Light Naphtha containing C5/C6 normal paraffin is favorable to the reactor as these form the branched-chain isomers which have higher octane numbers. Benzene-containing feeds can also be considered as feedstock. Feed Characterization for the ISOM unit is very crucial for a refiner and it is done using X-factor. It is defined as the sum of weight percentages of methylcyclopentane, cyclohexane, benzene, and C7+ hydrocarbons. X-factor ranges from 15 to 20 for a typical feed (Fathy et al., 2019). The typical composition of Light Naphtha is shown in Table 2.1. It is observed that most of them are C5/C6 components and a few C7+ components are present. The Isomerate composition from Table 2.1 shows that most of them are dominated by higher octane numbered components.

Table 2.1: Typical inlet and outlet composition of an ISOM reactor (Garg, class notes)

Component	%wt Light Naphtha	%wt Isomerate	Octane number
C4-	0.4	1.8	98
Isopentane	21.6	34.9	92.3
n-pentane	26.5	14.0	61.7
Cyclopentane	1.4	1.4	101.6
2,2DMB	0.9	13.4	91.8
2,3DMB	2.2	4.6	100.5
2-methylpentane	13.1	13.7	73.4
3-methylpentane	10.2	7.8	74.5
n-hexane	18.6	5.1	24.8
Methylcyclopentane	2.8	0.6	81.7
Cyclohexane	0.4	1.4	79.6
Benzene	1.9	0	101
C7+	0	0.3	0

2.2 Chemistry

The number of components involved in this process are 17. Hence, the reactions involved are higher in number. These high numbered reactions can be classified into 4 types of reactions:- Isomerization, Benzene saturation, Ring-opening of naphthenes, and hydrocracking reactions. All of them are discussed in detail in the following sections.

2.2.1 Isomerization Reaction

Isomerization is the chemical process by which a compound is transformed into any of its isomeric forms, i.e., forms with the same chemical composition but with different structures or configurations and, hence, generally with different physical and chemical properties. It is a desired reaction in the reactor and the reaction is slightly exothermic. An example shown in equation 2.1 is the conversion of pentane, a hydrocarbon with five carbon atoms joined in a straight chain, to its branched-chain isomer, isopentane.

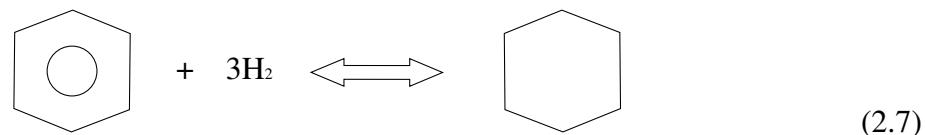


The mechanism for this reaction when a bi-functional catalyst pt-alumina is shown from equation 2.2 to equation-2.6 (Chekantsev et al., 2014). It is observed that formation of the carbocation is essential. Most of the refinery reactions form carbocation, even in the catalytic reformer.



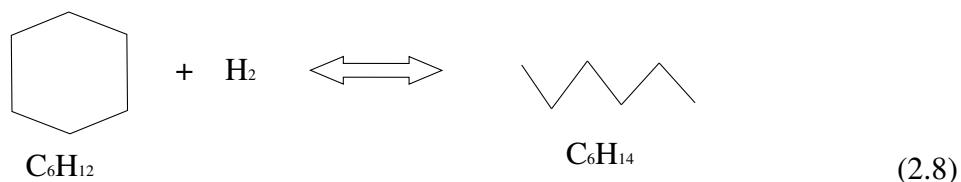
2.2.2 Benzene Saturation

Benzene is an undesired component in the gasoline pool hence, Benzene saturation is a desired reaction. Excess of hydrogen is required to saturate it. This reaction is highly exothermic. This reaction occurs over acidic sites (Valavarasu and Sairam, 2013). Here in the example-2.7 it is shown that benzene is converted into cyclohexane.



2.2.3 Ring Opening of Naphthenes

The Naphthenes obtained during the Benzene saturation or Naphthenes present in the feed gets involved in this reaction type. These components adsorb on the catalyst sites and inhibit the isomerization reactions. An example is shown in the equation-2.8



2.2.4 Hydro Cracking Reactions

This is the most undesirable reaction and favored during high temperature feed. These reactions lead to the loss of yield by forming gaseous components. An example of hydrocracking is shown in the equation-2.9



2.3 Thermodynamics

Thermodynamics is considered to be a limitation of the yield of an ISOM reactor. From the section-2.2.1 it is found that the isomerization reactions are reversible and hence Le-Chartlier's principles are applicable. The equilibrium of a reaction depends upon the temperature, pressure, and composition. These conditions should be applied theoretically:

- Since the reaction is slightly exothermic, i.e. $\Delta H_r = -4$ to -20 kJ/mol (Valavarasu and Sairam, 2013), reactor should be maintained at lower temperatures. Higher rates at a lower temperature can be attained by using pt on alumina type of catalyst.
- In the case of reaction-2.1 forward reaction must be higher than backward so, the concentration of nC_5H_12 should be higher than iC_5H_12 . This will be discussed in detail in later sections.
- Pressure maintained doesn't matter theoretically but it will be maintained higher due to 2 reasons: 1) volume of the reactor will be less and 2) Coke formation is reduced.

From Figure 2.1 yield of 2,2 DMB is decreasing as the temperature is increasing and the yield of 2MP and 3MP is increasing. Please also note that lower temperatures cannot be maintained as the catalyst is not activated at those temperatures. The figure also shows the temperature maintained for different catalysts.

2.4 Catalysis

ISOM catalyst is dual functional, they have both acidic and metallic sites. As seen from the equation-2.2 metallic sites help in hydrogenating or dehydrogenate reactions. Acidic sites help in catalyze isomerization reactions. Though acidic sites can perform isomerization reactions due to their high initial activity, a metal component is added to stabilize the reactions (Valavarasu and Sairam, 2013). The choice of catalyst depends upon the refiner if they want to increase the yield, they can use Chloride alumina or if they want to run their catalyst for longer periods, they

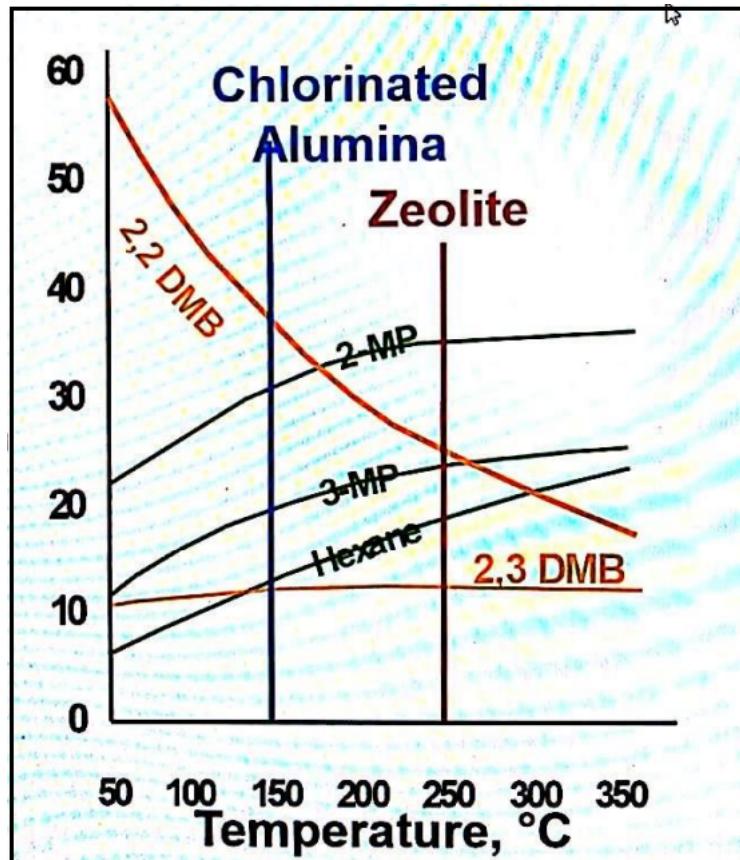


Figure 2.1: Effect of yield on C6 components upon increasing the temperature (Garg-class notes).

can use zeolites. An engineer at work can choose catalysts based on their operating conditions. The summary of different catalysts is presented in Table 2.2.

2.5 Operating Conditions

The extent of n-paraffin Isomerization reaction during the Isomerization process depends on the process variables, such as reactor temperature, operating pressure, space velocity, and H₂/HC ratio. The reactor temperature and LHSV¹ are the two important process variables.

Isomerization of Light Naphtha fraction. Reactor operating pressure and H₂/HC ratio do not have a significant effect on the conversion of n-paraffins to iso-paraffins under normal operating conditions. Industrial Isomerization units are normally operated at a molar hydrogen-

¹It is defined as the ratio of volumetric flow-rate of feed entering a reactor to the volume of the reactor

Table 2.2: Summary of Catalysts used in an Isomerization reactor (Valavarasu and Sairam, 2013)

Catalyst	Advantages	Disadvantages
Chloride alumina	Highest RON, Highest Activity, Highest yield.	Continuous Chloride addition is necessary, Sensitive to poisons.
Zeolite type	Regenerable, Highest stability, Can tolerate poisons	Lowest activity
Sulfated Zirconia	Posses intermediate activity, Tolerant to poisons, Regenerable	Requires higher H ₂ /HC ratio.

to hydrocarbon ratio of 0.05 mol/mol at the exit of the second reactor (Graeme and van der Laan, 2003). Typical operating conditions of different Isomerization processes are given in Table 2.3. The reason for choosing the temperature for catalyst chlorinated alumina is because of the extent of reaction of an isomerization reaction at higher temperatures. Its extent decreases because the catalyst just increases the rate of the reaction but not equilibrium. And for zeolites, this temperature is chosen because zeolites become active at this temperature and the extent doesn't matter because recycle mode is chosen for this kind of scenario and it doesn't affect the octane number from Figure 2.2. The effect of temperature on recycling is shown in Figure 2.2. It is observed that temperature has almost no effect on octane number in the case of recycling. In this case, one can use a zeolite type of catalyst for higher stability.

2.6 Performance Numbers

This section deals with some of the performance numbers that are dealt with in this context. These help a refiner in understanding how effective the Isomerization process is.

Table 2.3: Typical operating conditions based of an ISOM reactor based on catalyst (Valavarasu and Sairam, 2013)

Process Parameter	Chlorinate Alumina	Zeolites	Sulfated Zirconia
Tempearature, ($^{\circ}\text{C}$)	130-150	250-280	180-210
Pressure, barg	15-35	15-35	15-35
LHSV, (hr^{-1})	1.0-3.0	1.0-3.0	1.0-3.0
H ₂ /HC (mole)	1.0-2.0	1.0-20	1.0-2.0

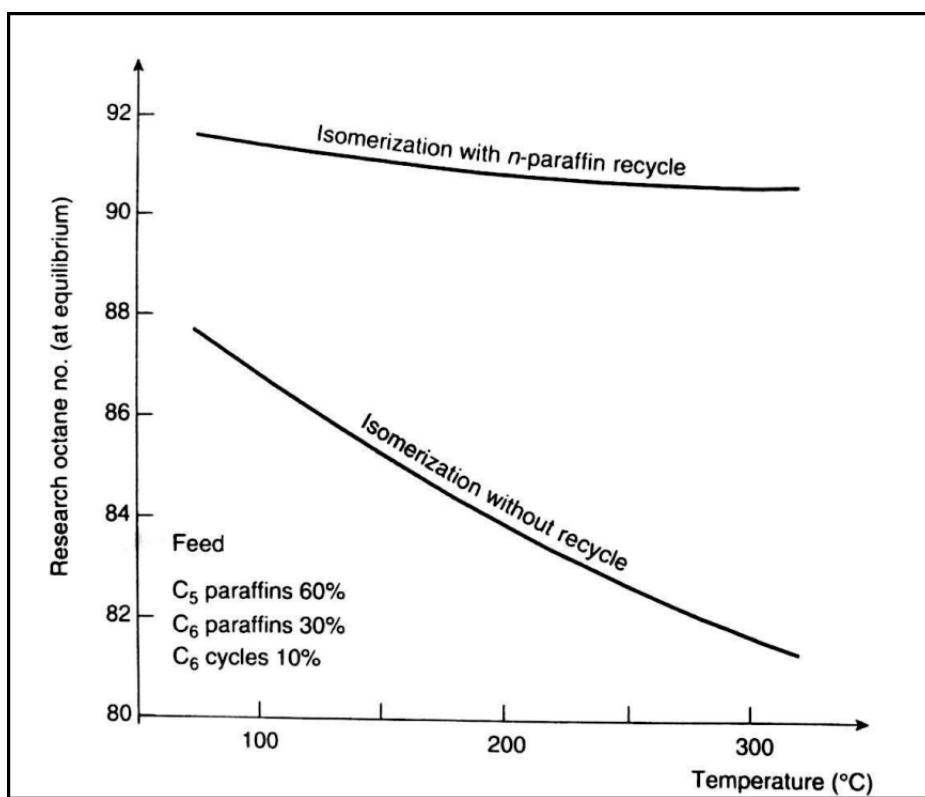


Figure 2.2: Effect of temperature with and without recycle on isomerate octane number (Garg-class notes).

2.6.1 Product Iso Ratios

Iso ratios are defined as the percentage of a component divided by the sum of their respective paraffin (Awan et al., 2021). There are 3 major ratios here. These components are important

because of their high octane numbers.

$$\frac{wt\% \text{ } iC5}{wt\% \text{ } iC5 + wt\% \text{ } nC5} \quad (2.10)$$

$$\frac{wt\% \text{ } 22DMB}{wt\% \text{ } nC6 + wt\% \text{ } 2MP + wt\% \text{ } 3MP + wt\% \text{ } 22DMB + wt\% \text{ } 33DMB} \quad (2.11)$$

$$\frac{wt\% \text{ } 23DMB}{wt\% \text{ } nC6 + wt\% \text{ } 2MP + wt\% \text{ } 3MP + wt\% \text{ } 22DMB + wt\% \text{ } 33DMB} \quad (2.12)$$

2.6.2 Paraffin Isomerization

Also called as PIN is defined as the sum of ratios mentioned in the equations-2.10, -2.11, -2.12. It is a measure of degree of Isomerization in the product (Awan et al., 2021).

2.6.3 Octane Number

Octane number is a standard measure of a fuel's ability to withstand compression in an internal combustion engine without detonating. The higher the octane number, the more compression the fuel can withstand before detonating. Iso-octane is used as a benchmark for the octane number. It has a value of 100, n-heptane has the octane number of 0. Two types of octane number are available: 1) Research Octane Number (*RON*) and 2) Motor Octane number (*MON*). *RON* is widely used and it is measured using a test engine under controlled conditions. It can also be calculated using the formulae given below. RON value of a typical Isomerate obtained from the Isomerization process is in the range of 83-90. *RON* is calculated using the equation given below.

$$RON = \sum_i y_i RON_i \quad (2.13)$$

2.7 Isomerization Processes

There are different flowsheets available in the ISOM unit, these flowsheets evolved with time in order to produce high yield and high octane numbered Isomerate. The flowsheets with their description are presented in the different sections below.

2.7.1 Once Through Process

The name suggests that the process feed is passed only once and no recycling of stream will be there, From the flowsheet 2.3 it is observed that 2 reactors are present. The first one is called lead, and the second is called a lag reactor. Both the reactors have almost the same configuration. The reason 2 of them present is that in the first reactor, complete benzene saturation, some hydrocracking reactions, and little Isomerization takes place because the amount of heat evolved is too much and Isomerization is not favored. In the second reactor, Isomerization reactions take place as the temperature of the outlet stream from the first reactor is reduced by placing a cooler. This stream is now sent to the stabilizer to separate the off-gases and the Isomerate stream. The usage of dryers before the reactor is to remove any water content present, because they act as poisons to the Chloride Alumina catalyst. UOP Penex, UOP Par-ISOM are some of the commercial processes.

2.7.2 Deisohexanizer process

A deisohexanizer is a multi component distillation column that helps in separating the iso-hexanes from n-hexane and n-pentane. From Figure 2.4 it is observed that recycling has a significant effect on the octane number and this idea is utilized here. The separated n-hexane and n-pentane is recycled back to the reactor, this helps in increasing the yield. Axen Hexorb Isomerization is one of the commercial processes that utilize this idea. In this, molecular sieves are used to desorb n-hexanes (Valavarasu and Sairam, 2013).

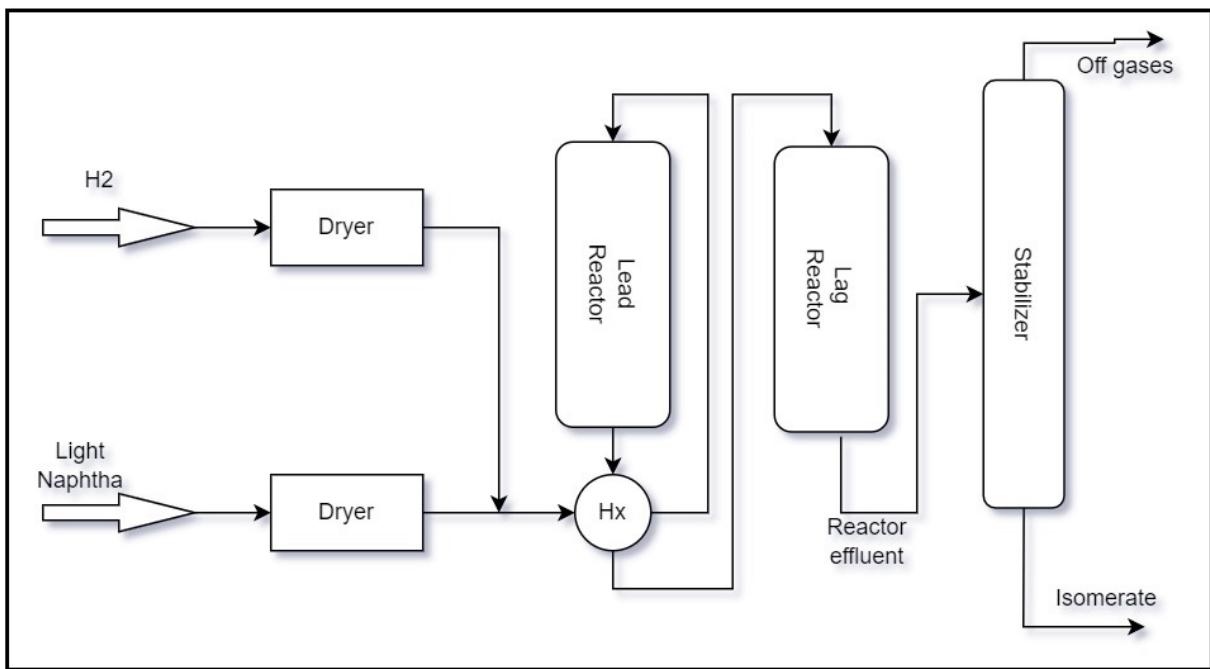


Figure 2.3: Flowsheet showing the Once Through process (Garg-class notes).

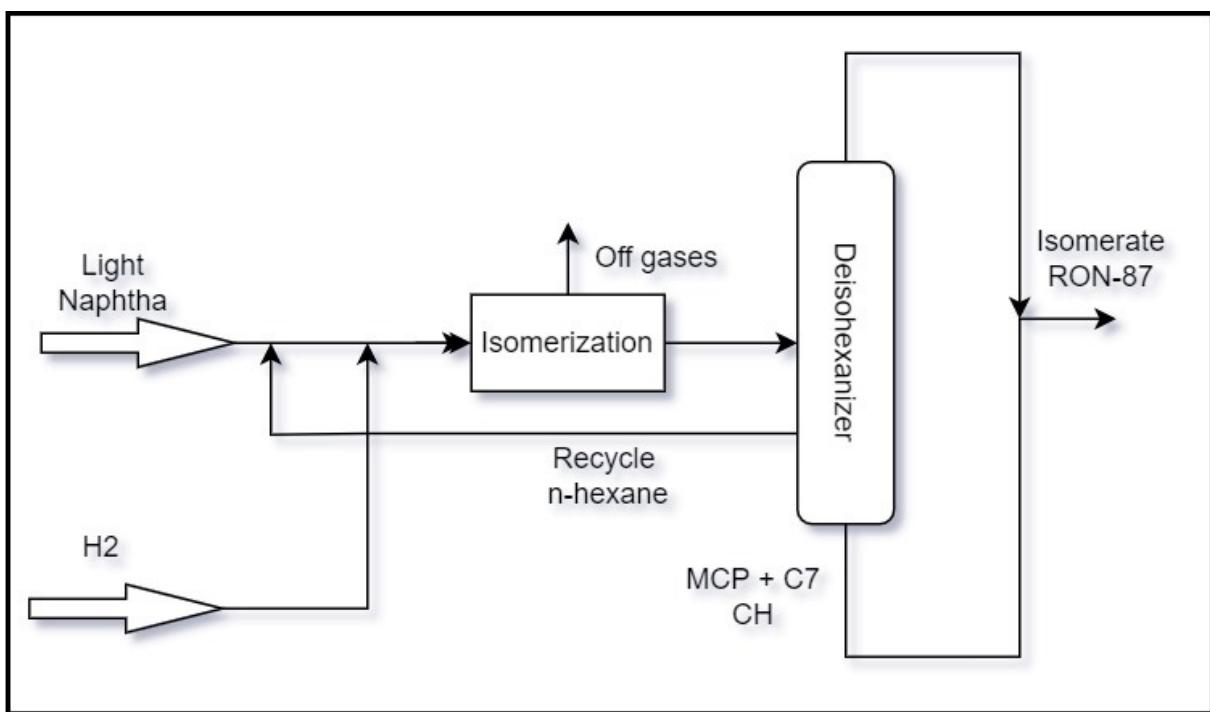


Figure 2.4: Isomerization unit containing the Deisohexanizer (Garg-class notes).

2.7.3 Deisopentanizer with Deisohexanizer process

The idea here is simple, as per the Le-Chartlier's principle if the composition of the product is less in a equilibrium reaction, the reaction proceeds forward. So, the removal of isopentane

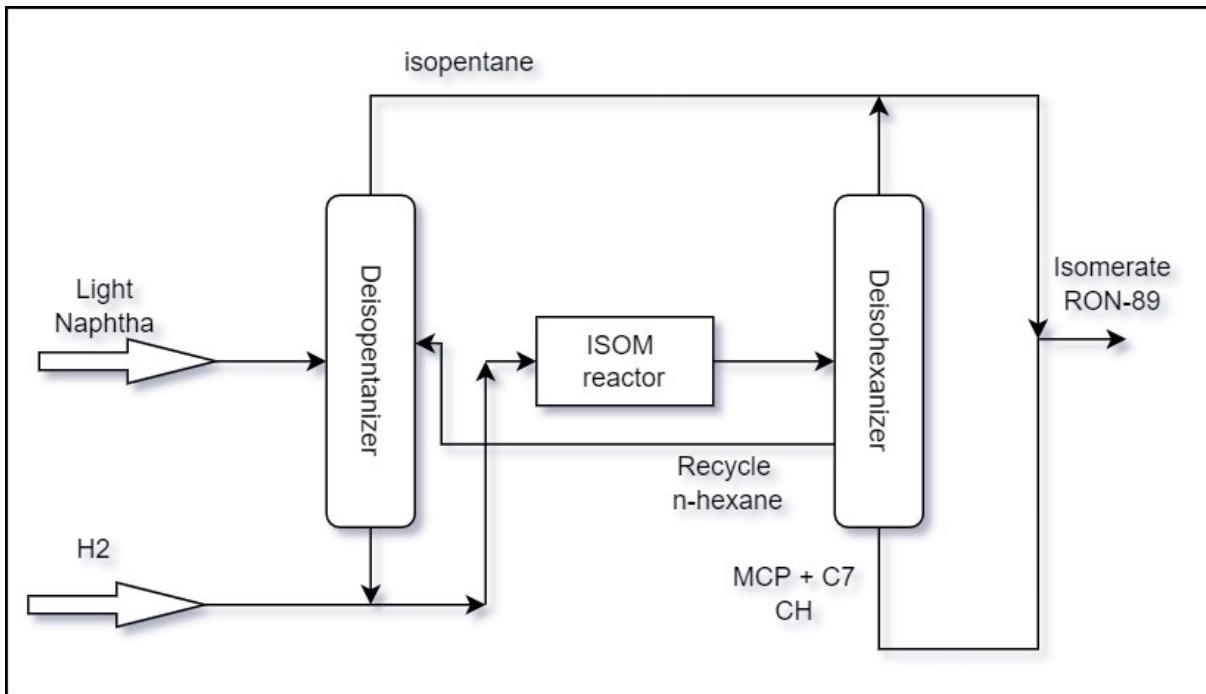


Figure 2.5: Isomerization unit installed with the Deisopentanizer and hte Deisohexanizer (Garg-class notes).

from the reaction 2.1 will improve the forward rate of reaction. Here isopentane present in the fresh feed stream was removed by deisopentanizer. Also, a deisohexanizer is present at the end of the reactor to remove unconverted n-hexane. Axen Ipsorb along with Axen Hexorb is used, GTC technology is some of the commercial processes. This configuration is mostly used among the refineries because this method gives the better yield. The typical flowsheet is presented in the Figure 2.5

This chapter explained every aspect of an Isomerization unit, from Feed Characterization to different configurations used. Finally, it depends upon the refiner in charge to choose which type of catalyst and which type of configuration based upon their plant economics.

Chapter 3

Vapor Phase Modeling of the Reactor

A universal model suggested by Chekantsev et al. (2014), is the basis for many of the models published. In the Ahmed et al. (2018), the order of the reaction is also taken as a parameter that is optimized. In Enikeeva et al. (2021), the reaction model contains 54 reactions which don't have anything related C7+ components, the reactions are shown in the Figure A.1 and the model is shown below. From this model, we can write the component i balance if in the reaction j .

3.1 Component Balance

The model of isomerization reactor is a plug flow model and it is given as:

$$\frac{dC_i}{dL} = \frac{ACS}{q} \sum_{j=1}^{j=54} r_j \quad (3.1)$$

$$r_j = k_j C_i C_{H_2} \quad (3.2)$$

$$k_j = k_{j_0} \exp \frac{-E_j}{RT} \quad (3.3)$$

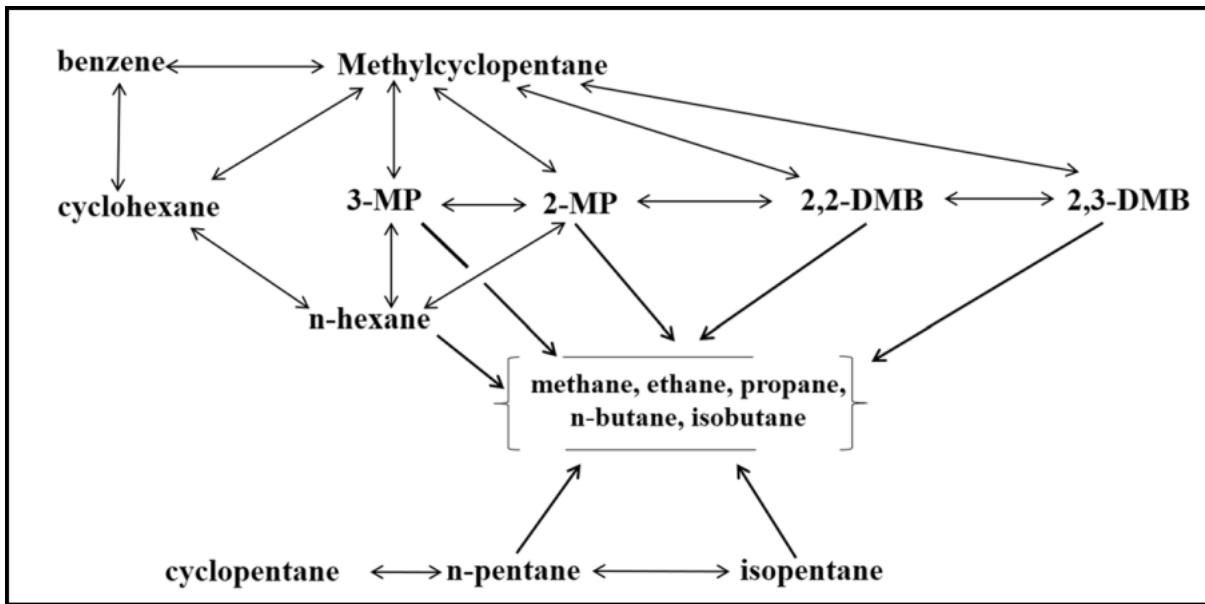


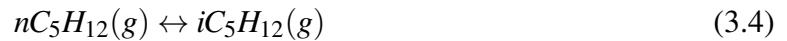
Figure 3.1: Reaction Model used, these reactions can be divided into 4 types of reactions as discussed in section 2.2 (Enikeeva et al., 2021)

Here L (m) represents the length of the reactor, ACS (m^2) represents the area of the cross section of the reactor, and q represents the volumetric flow rate of the feed entering the reactor. r_j ($mol/m^3 - hr$) represents the rate of the component formation or depletion in the j^{th} reaction. The sign of r_j is + when the component is on the product side and – when the component is on the reactant side. The value of a in equation 3.2 is 0 in the case of Isomerization reaction and in the rest of the cases, it is 1. The no. of components are 17 and is shown in Table 3.4. From the component mole balance, we get a total of 17 ODEs because 17 components are present in our system.

3.2 The Importance of Rate Constant (k_j)

The k_j in the equation 3.2 encapsulates all the mass transfer resistances. The Isomerization unit is operated under different operating conditions in different plants. Some of them operate the reactions in a liquid phase and some of them in a vapor phase. If the temperature of the inlet stream is greater than $190^\circ C$ then the reactions happen in the vapor phase or the amount of hydrogen used is more, then the reactions happen in the vapor phase. If the stream entering the

reactor is around $130^{\circ}C - 160^{\circ}C$ then the reactions happen in the liquid phase.

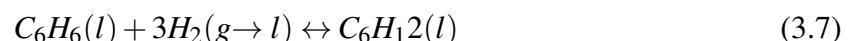


Consider an isomerization reaction 3.4 of order 1 is taking place in the vapor phase upon the catalyst. Then the equation 3.2 becomes equation 3.6.

$$r_j = \eta k_j C_{nC_5H_{12}} \quad (3.5)$$

$$r_j = K_j C_{nC_5H_{12}} \quad (3.6)$$

where η is the effectiveness factor of the catalyst.



Consider a reaction 3.7 happening in the liquid phase where the hydrogen present in the gaseous phase will enter the liquid phase. In general, the solubility of hydrogen in Light Naphtha decides the rate of reaction. Then the rate equation 3.2 will become as equation 3.10.

$$r_j = k_j C_{C_6H_6} C_{H_2}^{liq} \quad (3.8)$$

$$r_j = k_j C_{C_6H_6} \left(H * C_{H_2}^{vap} \right) \quad (3.9)$$

$$r_j = K_j C_{C_6H_6} C_{H_2}^{vap} \quad (3.10)$$

The modified rate equations are now tuned by the optimizer based on the data. So, one need not worry if the reaction happens in the liquid phase or vapor phase and also about the calculation of the effectiveness factor, Henry's constant, etc.

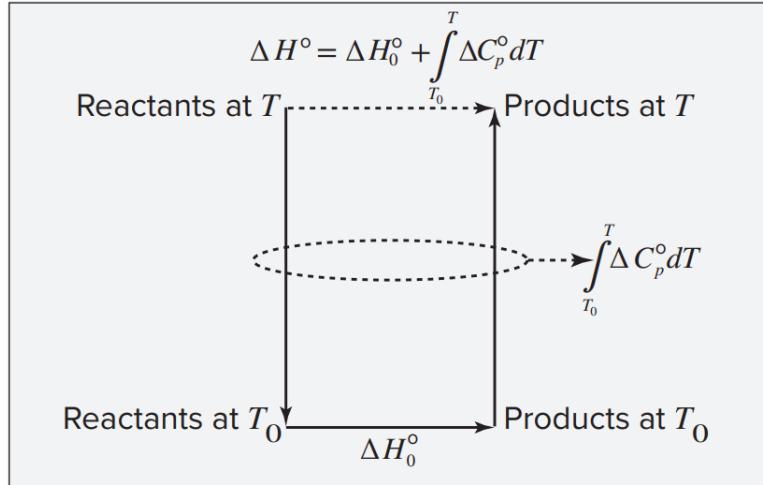


Figure 3.2: Figure showing the breakdown of how heat of reaction is calculated at elevated temperatures(Hess Law) (Smith et al., 2005)

3.3 Energy Balance

The energy balance for the PFR model is given as:

$$\frac{dT}{dL} = -\frac{ACS}{q} \frac{\sum_{j=1}^{54} Q_j r_j}{Ca_m C p_m} \quad (3.11)$$

Here T represents the Temperature(K) across the reactor length, Q_j (kJ/mol) represents the heat evolved during a reaction j , Ca_m represents the overall concentration of the reaction mixture and Cp_m represents the heat capacity of the reaction mixture. One ODE from the energy balance is obtained. A total of 18 ODEs are solved simultaneously with some Initial values using OpenModelica solver.

$$Q_j = \Delta H_R + \Delta H_0^\circ + \Delta H_P \quad (3.12)$$

Where ΔH_R is the amount of heat required to cool the reaction mixture to a standard temperature. ΔH_P is the amount of heat required to heat the product mixture to the reaction temperature T . From the Figure 3.2 it is shown $\int_{T_0}^T \Delta C_p^\circ dT$ as the sum of ΔH_R and ΔH_P . ΔH_0° is the heat of reaction at the standard state.

3.4 Thermodynamic Modeling

ΔH_P , ΔH_R and ΔC_{p_m} from the equation 3.12 is given by the equation 3.13, the real property is the sum of ideal and residual property.

$$\Delta H_P = \Delta H_P^{id} + \Delta H_P^{res} \quad (3.13)$$

$$\Delta H_R = \Delta H_R^{id} + \Delta H_R^{res} \quad (3.14)$$

$$C_{p_m} = C_{p_m}^{id} + C_{p_m}^{res} \quad (3.15)$$

To calculate residual properties, the PengRobinson Equation of State is used here. Compressibility factor calculation for the individual component at a given temperature and pressure is done initially, the formula is given by equation 3.17.

$$Z^3 + (B - 1)Z^2 + (A - 3B^2 - 2B)Z + (B^3 + B^2 - AB) = 0 \quad (3.16)$$

$$A = \frac{aP}{RT^2} \quad (3.17)$$

$$B = \frac{bP}{RT} \quad (3.18)$$

$$a = 1 + c(1 - T_r^0)^2 \times 0.45724 \frac{RT_c^2}{P_c} \quad (3.19)$$

$$b = 0.0778 \frac{RT_c}{P_c} \quad (3.20)$$

$$c = 0.37464 + 1.54226\omega - 0.26992\omega^2 \quad (3.21)$$

Similarly, Z_m is calculated using the equation 3.17 but replacing B with B_m and A with A_m , and those values are calculated using the equation 3.23.

$$A_m = \frac{a_m P}{RT^2} \quad (3.22)$$

$$B_m = \frac{b_m P}{RT} \quad (3.23)$$

$$a_m = \sum_i \sum_j y_i y_j \sqrt{a_i a_j} (1 - k_{ij}) \quad (3.24)$$

$$b_m = \sum_i y_i b_i \quad (3.25)$$

The Z for each component obtained from equation 3.17 and Zm are used to calculate molar volume using equation 3.26 and equation 3.27

$$V = \frac{RTZ}{P} \quad (3.26)$$

$$V_m = \frac{RTZ_m}{P} \quad (3.27)$$

The calculation of Cp^{res} for the equation 3.15 is given by equation 3.28 and Cp^{id} is given by the equation 3.29. Cp^{res} is an average property hence it uses variables like b_m and V_m .

$$C_{p_m}^{res} = -R + T \frac{dP}{dT} \frac{dV}{dT} - T \frac{d^2a/dT}{\sqrt{8}b_m} \log \left(\frac{V_m + (1 - \sqrt{2})b_m}{V_m + (1 + \sqrt{2})b_m} \right) \quad (3.28)$$

$$C_{p_m}^{id} = \sum_{i=1}^{17} y_i C_{pi}^{id} \quad (3.29)$$

$$C_{pi}^{id} = A + BT + CT^2 + \frac{D}{T^3} \quad (3.30)$$

The calculation of ΔH^{res} for the equation 3.13 is given by equation 3.31. ΔH^{res} is calculated for every reaction and the values like V, Z are utilized for the components involved in those reactions.

$$\Delta H^{res} = -R + T \frac{dP}{dT} \frac{dV}{dT} - T \frac{\frac{d^2a}{dT}}{\sqrt{8}b} \log \left(\frac{v + (1 - \sqrt{2})b}{v + (1 + \sqrt{2})b} \right) \quad (3.31)$$

All these equations related to Peng Robinson are taken from the github repo of DWSIM. The code written in Modelica language is provided in the section A.1.

3.5 OpenModelica

OpenModelica is an open source software based on Modelcia language. Modelica is primarily a modeling language to specify even a complex mathematical model easily. It is inter-disciplinary i.e., models developed in Electrical, Mechanical disciplines can be easily used together to build a unique model. Modelica is also an object-oriented, equation-based programming language which is devoted towards solving complex problems that requires high performance (Fritzson,

2013). Important features of OpenModelica are:

Acasual Modeling: In Modelica we can define the model equations without worrying about the algorithm. For instance if we want to solve a flash problem in Matlab we need to provide a structure flow to solve for the problem, whereas in the Modelica we all need to provide is the modelling equations.

Object Oriented: Modelica is an object-oriented language with a general class concept. This helps in the reuse of models, it is as simple as drag and drop objects.

Multi domain Modeling Capability: Components corresponding to physical objects from several different domains such as electrical, mechanical, thermodynamic, hydraulic and control applications.

Graphical: Modelica has a strong software component model, with constructs for creating and connecting components. Users can build the models and reuse them for creating system by connecting them. Example here is we can build the models like material stream, reactors, separators and connect them to create a flowsheet.

OpenModelica's ability to solve the complex ODE's effectively and its Acasual modeling are the main reasons to choose this open source software. DASSL solver is used in default in the OpenModelica. DASSL is a Differential Algebraic Equation (DAE) Solver , which can solve DAE's of Index¹-2 easily. Since ODE is an Index-0 problem, DASSL is able to solve.

¹Index of a DAE is defined as the number of derivatives required to convert a DAE into ODE.

Table 3.1: Table showing the Design parameters used in the vapor phase model

S.No.	Design parameter	value
1	weight of catalyst (kg)	27000
2	density of catalyst (kg/m^3)	2500
3	Area of cross section of reactor (m^2)	4.9
4	length of the reactor (m)	2.19
5	void fraction (ϕ)	0.5

Table 3.2: Table showing the operating conditions of material stream used in the Vapor phase model(Enikeeva et al., 2021)

S.No.	Operating parameter	value
1	Temperature ($^{\circ}C$)	147
2	Pressure (kPa)	3200
3	Flow rate ($kmol/hr$)	5653

3.6 Simulation Results

The data for running this model is taken from the Enikeeva et al. (2021). It is not mentioned in which phase the reaction mixture is. Peng Robinson's model in Aspen-Hysys indicated it as the vapor phase for its respective composition and operating conditions. All the 54 reactions mentioned in Figure A.1 are considered. Note that this does not contain the reactions related to C7+ components as these components are hardly present in the feed.

Operating conditions and design parameters are mentioned in the Table 3.2, Table 3.1 respectively. The kinetic parameters considered are shown in Table 3.3. The composition of the inlet feed is shown in Table 3.4. The data we are considering is having 3 reactors each of 9000 kg catalyst. A cooler is installed after reactor-2, which cools the effluent to $147^{\circ}C$. The density of the catalyst is not mentioned in the paper, hence we assumed $2500kg/m^3$ and a void fraction of 0.5 is considered. With the assumption of the diameter of the reactor as 2.5 m. The length of the reactor came out to be 2.19 meters. The flowsheet is presented in the Figure 3.3 All the results will be shown along the reactor length. The concentration profiles of the key components

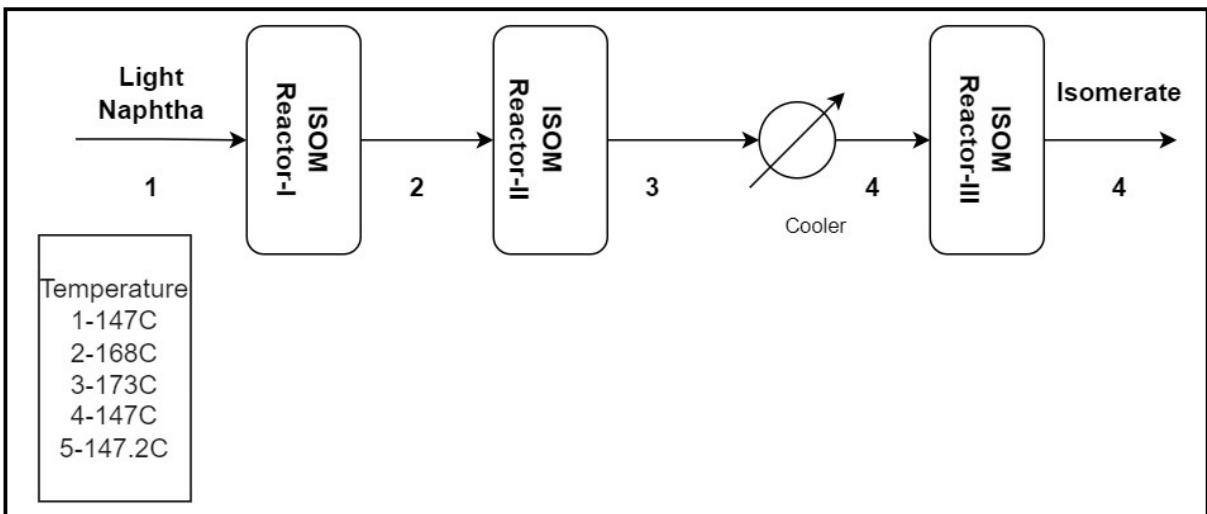


Figure 3.3: Figure showing the schema of reactors arranged in Enikeeva et al. (2021)

are shown in figures 3.4, 3.5, 3.6 and 3.7. A bar graph comparing the composition of outlet of the reactor is shown in Figure 3.8. In the current work we have modeled ISOM reactor as one reactor equivalent to 3 reactors of Enikeeva et al. (2021). The inter-cooler location corresponds to 1.46 meters at the current work. So the temperature value of the reactor is fixed at 147($^{\circ}$ C), which is same as mentioned in the Enikeeva et al. (2021) after inter-cooler.

The kinetic parameters considered to solve the model are fitted to the model in Enikeeva et al. (2021). Since the model developed in this work is different from the Enikeeva et al. (2021) the results are not matching with the literature data. Tuning of kinetic parameters will be done in the later chapters. This completes the discussion on this chapter.

Table 3.3: Table showing the initial kinetic parameters for the vapor phase model (Enikeeva et al., 2021)

rxn	k0 (1/hr)	Ej (kJ/mol)	rxn	k0 (1/hr)	Ej (kJ/mo)
1	2.7477E+19	148.93	28	1.02087E+12	129.29
2	4.15882E+19	154.28	29	3.62218E+18	154.54
3	1.94523E+11	143.17	30	7.56781E+13	98.63
4	2.39315E+11	151.41	31	3.15479E+17	150.29
5	6.01132E+21	150.98	32	48861861048	102.35
6	1.69422E+22	155.92	33	1.14543E+15	168.13
7	7.74408E+19	152.96	34	50000000000	90.7
8	7.74408E+19	149.95	35	3.88124E+24	177.32
9	1.44202E+15	127.28	36	3.53973E+14	222.9
10	4.06415E+16	139.07	37	2.56431E+16	59.8
11	574076810.7	64.5	38	1.61797E+15	54.08
12	46662715040	77.06	39	1.50998E+14	330.28
13	2.39315E+14	146.14	40	7.56781E+28	329.06
14	3.30347E+15	160.28	41	7.39554E+33	284.97
15	2.23342E+13	98.28	42	8.689E+13	166.32
16	2.23342E+14	105.4	43	9.97631E+13	165.82
17	12559432.16	3.51	44	3.15479E+13	112.05
18	42556901.91	4.79	45	3.53973E+28	265
19	1.4756E+34	180.2	46	3.53973E+16	264
20	5E+26	400.43	47	3.53973E+28	263.5
21	1.44202E+30	187.05	48	4.25569E+30	265
22	5E+28	300.79	49	2.23342E+34	295.62
23	7294071.301	51.08	50	147560461.3	295
24	3154786.722	341.89	51	397164117.4	295.19
25	3.70655E+13	129.75	52	125594321.6	294
26	33804148770	88.64	53	3.38041E+31	294.22
27	7.39554E+16	135.45	54	3.45915E+23	278.81

Table 3.4: Table showing the list of components and their composition used in the vapor phase model (Enikeeva et al., 2021)

S.No.	Component	Mass Fraction	Mole Fraction
1	n-Pentane	0.161843	0.0344
2	i-Pentane	5.54E-02	0.01178
3	n-Hexane	0.168385	0.0299
4	2-Mpentane	0.186026	0.0331
5	3-Mpentane	0.145986	0.0259
6	22-Mbutane	6.94E-03	0.0123
7	23-Mbutane	3.75E-02	0.0067
8	Benzene	1.05E-02	0.00148
9	Cyclohexane	8.13E-03	0.0020
10	Hydrogen	0.105748	0.8048
11	Mcyclopentan	5.17E-02	0.0091
12	Cyclopentane	1.10E-02	0.0024
13	n-Butane	9.91E-04	0.0003
14	i-Butane	3.47E-03	0.001
15	Propane	6.14E-03	0.0021
16	Ethane	1.16E-02	0.0059
17	Methane	2.86E-02	0.0027

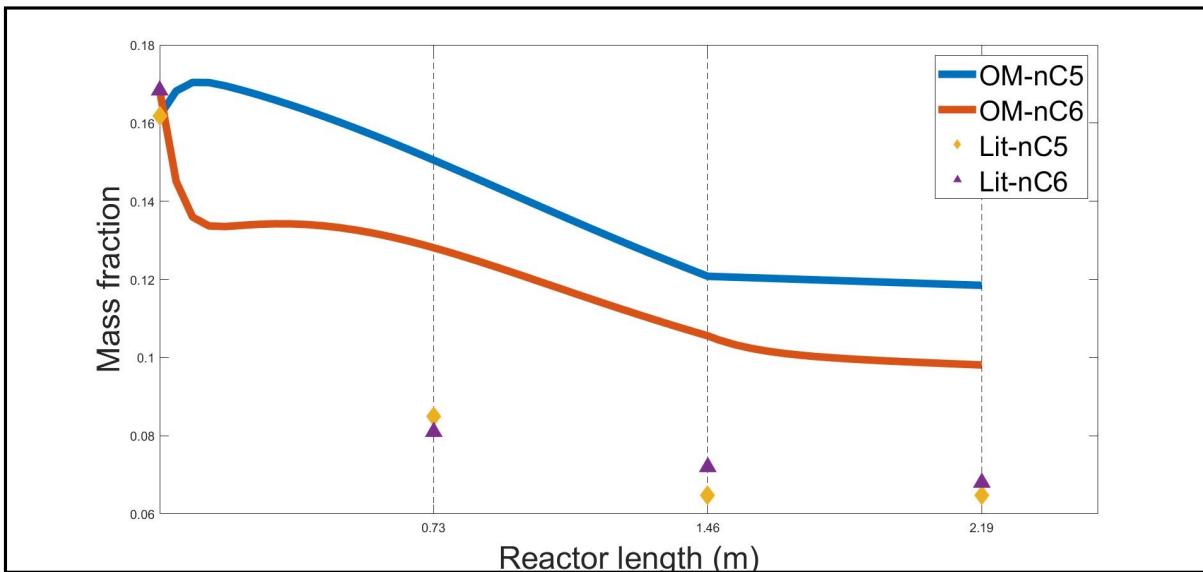


Figure 3.4: Figure showing the plots of n-paraffins of across the reactor length. The solid lines represent the simulation results and the points represent the Enikeeva et al. (2021) data. Vertical grid lines showing the reactor end.

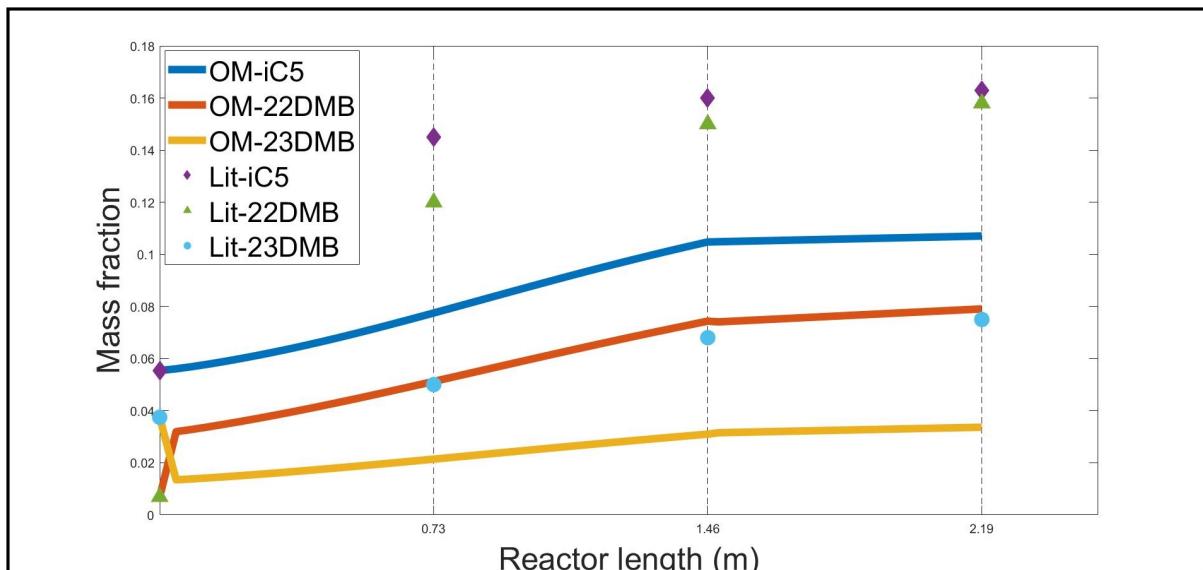


Figure 3.5: Figure showing the plots of value added components across the reactor length. The solid lines represent the simulation results and the points represent the Enikeeva et al. (2021) data. Vertical grid lines showing the reactor end.

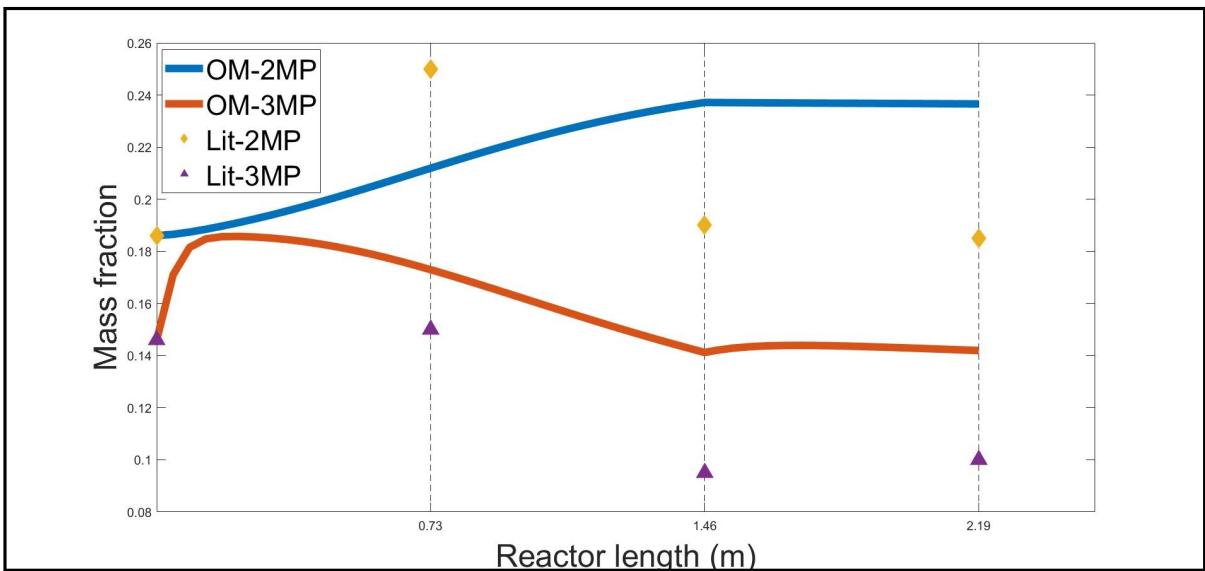


Figure 3.6: Figure showing the plots of 2MP and 3MP weight fraction across the reactor length. The deviation is due to the different model considered here. The solid lines represent the simulation results and the points represent the Enikeeva et al. (2021) data. Vertical grid lines showing the reactor end.

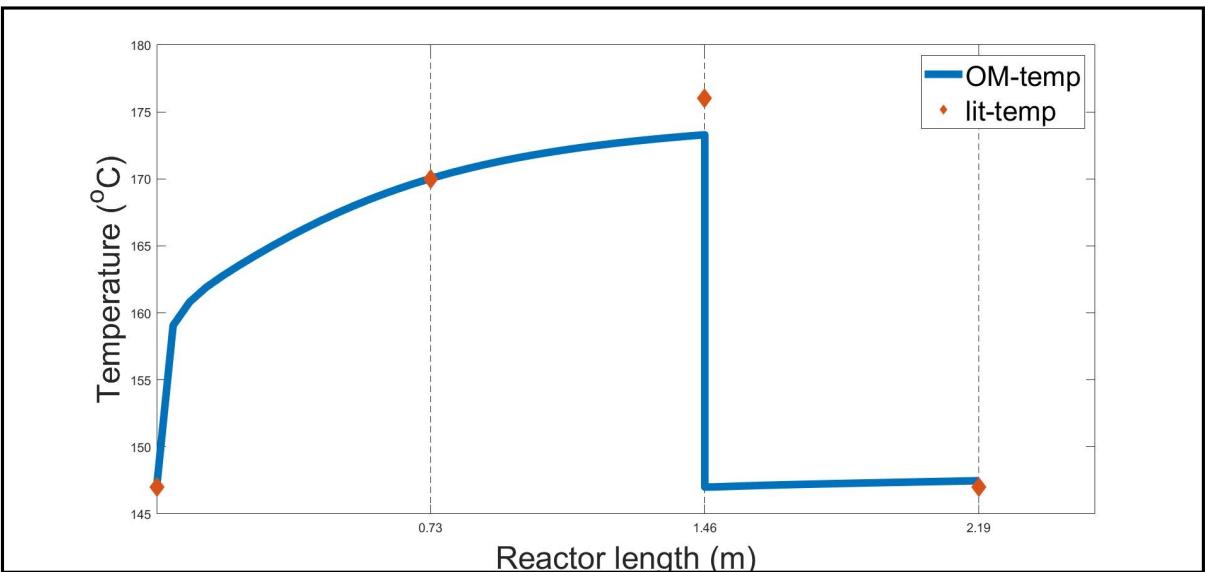


Figure 3.7: Figure showing the comparison plots of temperature profile across the reactor. The results are not matching with the Enikeeva et al. (2021) data because we have different model considered in this work. The sudden drop in the temperature profile at 1.46m is due to the cooler. The solid lines represent the simulation results and the points represent the Enikeeva et al. (2021) data.

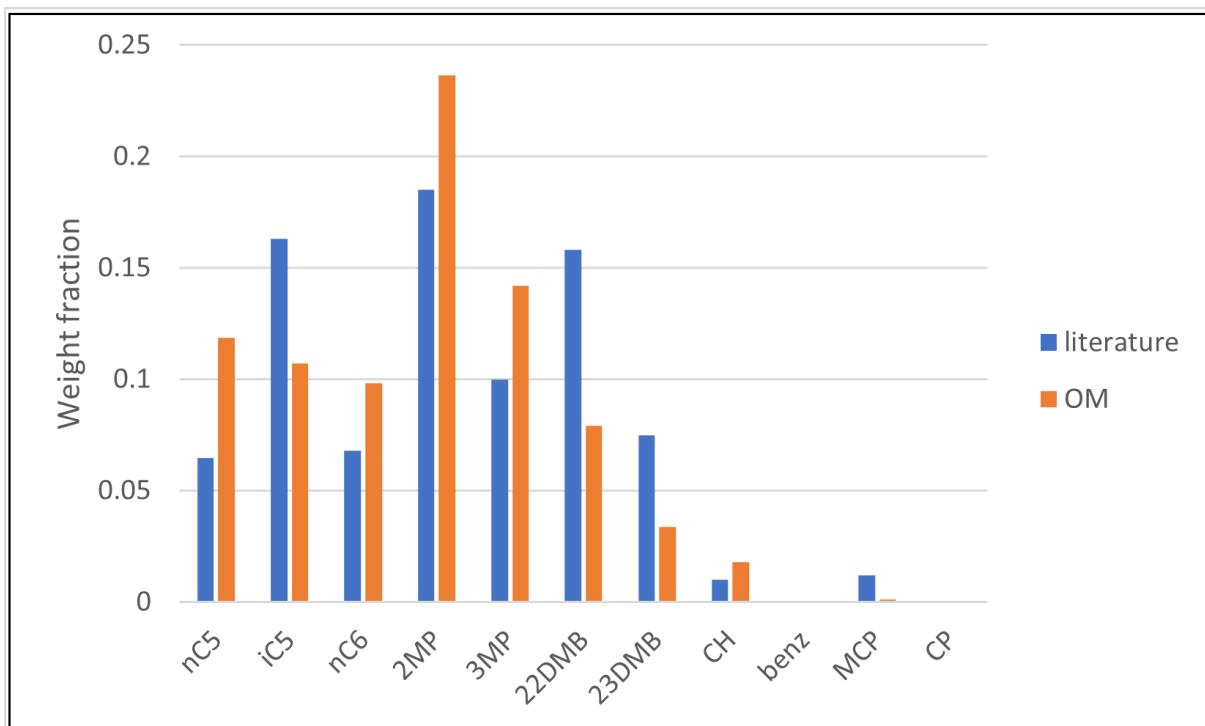


Figure 3.8: Bar graph showing the comparison of outlet weight fraction between literature and OM model.

Chapter 4

Function based Optimizer in OpenModelica

The optimizer helps to fit the given data to the proposed model by tuning the parameters. The parameters tuned here in this model are the activation energies and the pre-exponential factors. Optimizers can be specified in two broad categories: 1) Derivative based and 2) Function evaluation based. Function evaluation based, COBYLA optimizer is chosen in this work. The importance of choosing a function-based optimizer is:

- In every optimization problem, there is an objective function that needs to be minimized. The objective function evaluation is done by the output of the simulation. As a result, for every objective function evaluation, there will be a simulation.
- Hence, the objective function may not be a smooth function of the tuned parameters, which in this case are independent variables.
- Since, the objective function may not be seen be smooth and the question of calculating the gradient doesn't even arise.

The work done by the typical optimizer is shown in Figure 4.1. The explanation is given here:

1. Initial guess for the parameter should be given. Then the optimizer calls the simulation, and the simulation runs and produces results. These results are read by the optimizer and calculate the objective function.
2. If the convergence criterion is reached, then the algorithm ends. And declares the parameters in hand as the optimal parameters.
3. Else calculates the new parameters and calls the model again, and this loop continues until the convergence criterion is reached.

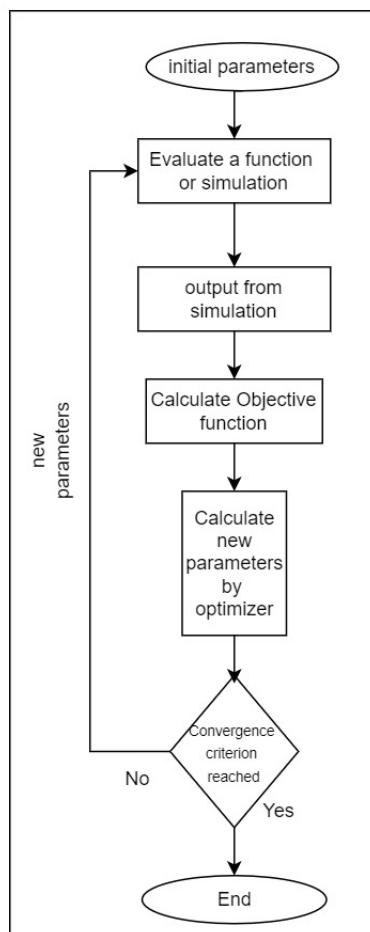


Figure 4.1: Typical algorithm to solve an optimization problem

4.1 COBYLA Optimizer

COBYLA is developed by Powell which stands for Constrained Optimization BY Linear Approximation. Before an understanding how COBYLA works, one needs to know about Nelder and Mead optimizer. Because, COBYLA is an extension of Nelder and mead. In each iteration, the optimizer calculates the value of the objective function at $n+1$ points, $\bar{x}^j : j = 0, 1, \dots, n$ say. These $n+1$ points form a convex hull. Let \bar{x}^l be one of the vertices of the convex hull, where the objective function value is maximum. Hence, this needs to be replaced by a new vertex using equation 4.1 (Nelder and Mead, 1965).

$$\bar{x}_{new}^l = -\theta \bar{x}^l + (1 + \theta)n^{-1} \sum_{j=0, j \neq l}^n \bar{x}^j \quad (4.1)$$

Where, θ is called reflection constant and lies in the interval $(0, 1)$. This step can be interpreted as the reflection of the point \bar{x}^l w.r.t. mean of the remaining points. If the iterations failed to generate a new small value, then the algorithm shrinks the convex hull to n points. COBYLA also updates the parameters based upon this method, except it has many constraints to tackle. It has the following features:

- If an optimization problem is defined as

$$\text{minimize } F(\bar{x}), \bar{x} \in \mathbb{R}^n \quad (4.2)$$

$$\text{subject to } C_i(\bar{x}) \geq 0, i = 1, 2, 3, \dots, m \quad (4.3)$$

- Then COBYLA approximates the problem 4.2 as problem-4.4 where the function $\hat{F}(\bar{x})$, $\hat{C}_i(\bar{x})$ interpolates the function $F(\bar{x})$, $C_i(\bar{x})$.

$$\text{minimize } \hat{F}(\bar{x}), \bar{x} \in \mathbb{R}^n \quad (4.4)$$

$$\text{subject to } \hat{C}_i(\bar{x}) \geq 0, i = 1, 2, 3, \dots, m \quad (4.5)$$

- A trust region is imposed to have some control over the steps taken by the algorithm.

This trust-region remains constant until predicted steps improve the objective function and feasibility conditions fail to occur.

- To compare the goodness of two vectors, a merit function is employed, given by equation 4.6. Where, subscript + represents that if none of them is positive, the value in place of brackets becomes 0. \bar{x} is better than \bar{y} only if $\phi(\bar{x}) \leq \phi(\bar{y})$. The algorithm is shown in Figure 4.2. For more details on how the algorithm works, refer Powell (1994).

$$\phi(\bar{x}) = F(\bar{x}) + \mu [max\{-C_i(\bar{x} : i = 1, 2, 3, \dots m)\}]_+ \quad (4.6)$$

4.2 Providing COBYLA Optimizer in OpenModelica through Interoperability

Integration of different software's is technically called Interoperability. In the figure-4.3 the rectangle box representing as optimizer is the algorithm written in R-language. And the box representing modelica is written in Open-Modelica. The optimizer package in R software should call the model of OpenModelica by passing the parameters. Then the model runs and produces output and this is read by R-language. This chapter explains these things technically. The most important thing here is that this delay due to data transfer is very much less when compared to run the simulation.

4.2.1 Some Basic Commands of OpenModelica

When an OpenModelica script file is compiled for the first time, it generates an executable file, XML files, object files, and CSV files. These files have their own usage. For example, the executable file is a compilation of the OM script file. It can be run repeatedly as if it is a model, the CSV file has an output results of the model. If the name of the file is ‘Model.mo’. Enter ‘cmd prompt’ where the file is present and type the command *omc Model.mo*. An example of this is shown below.

C:\users\pavan\documents\openModelica>omc Model.mo

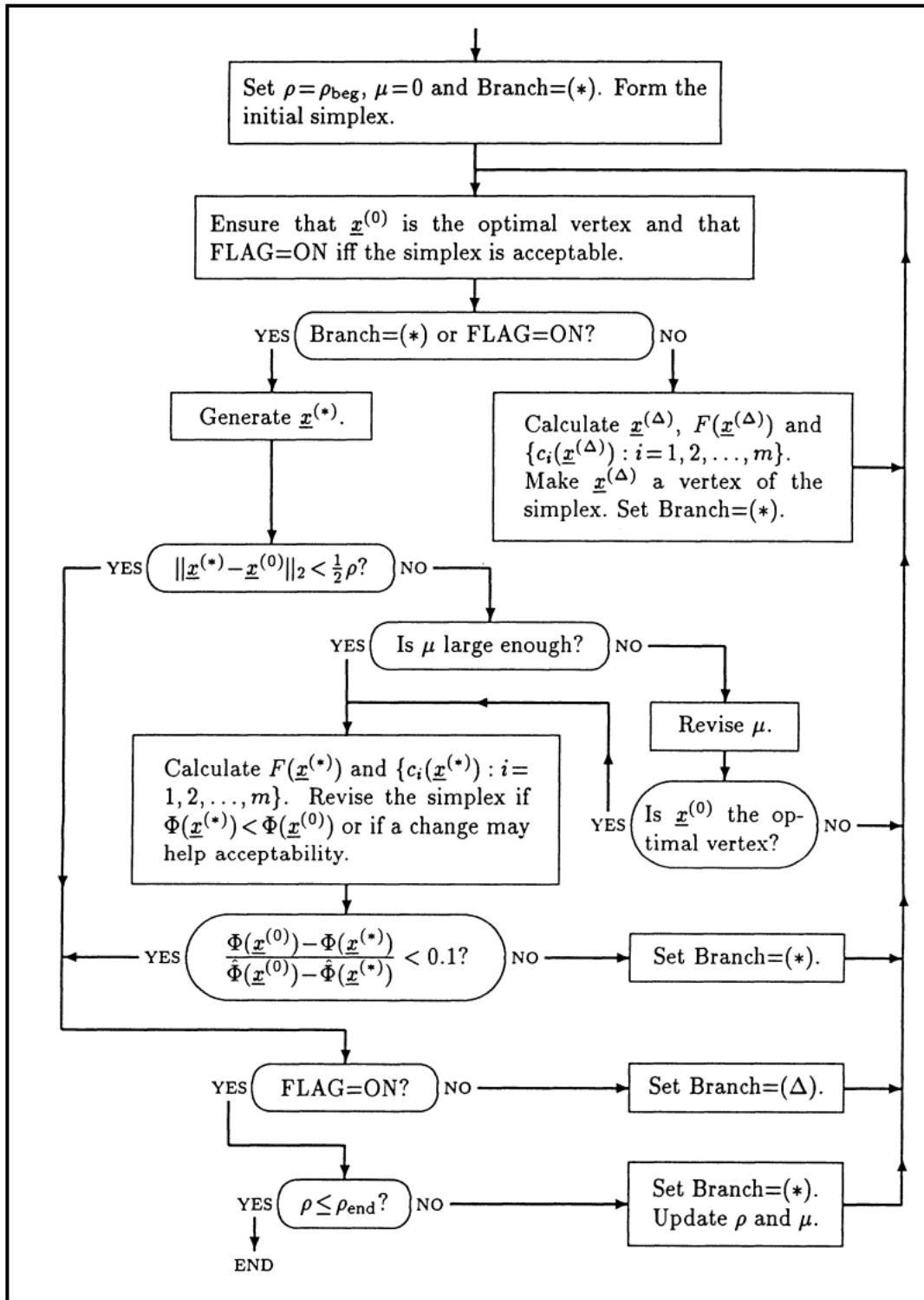


Figure 4.2: Algorithm of the COBYLA optimizer (Powell, 1994)

This command compiles the OpenModelica model for the first time and this takes 10-15 seconds based upon complexity of the problem. This command generates all the files mentioned above. Now if one wants to change the parameters of the ‘Model.mo’ say, par_1 no need to edit the script file and run the model again. OpenModelica has a facility to change the parameters of the model through the executable file produced. It can be done in this way:

```
C:\users\pavan\documents\openModelica>Model.exe -override=par_1=10
```

Parameter of the ‘Model.mo’ as in the parameter of a mathematical model, in our case it is kinetic parameters we want to change repeatedly. This method will take few milliseconds to run the executable with the updated parameters. The results get updated immediately in the same CSV file. This idea is utilized to run the optimization algorithm in *R*.

4.2.2 Interoperability of R Software and OpenModelica

R language *system* command is used to pass the command line arguments. *system* a built-in function in *R* will be able to pass the arguments. For instance *system(paste0("Model.exe"))* will run the OpenModelica model once through the command line. Since the optimizer needs to evaluate the objective function in every iteration, this *system* command helps in passing the command line arguments from the *R* whenever it wants. Function *read.csv()* will read a CSV file from the *R*. The workflow will now be changed from Figure 4.1 to Figure 4.3. This Figure 4.3 gives a fair understanding of how interoperability works. The code for this optimization algorithm is given in the section-A.2.

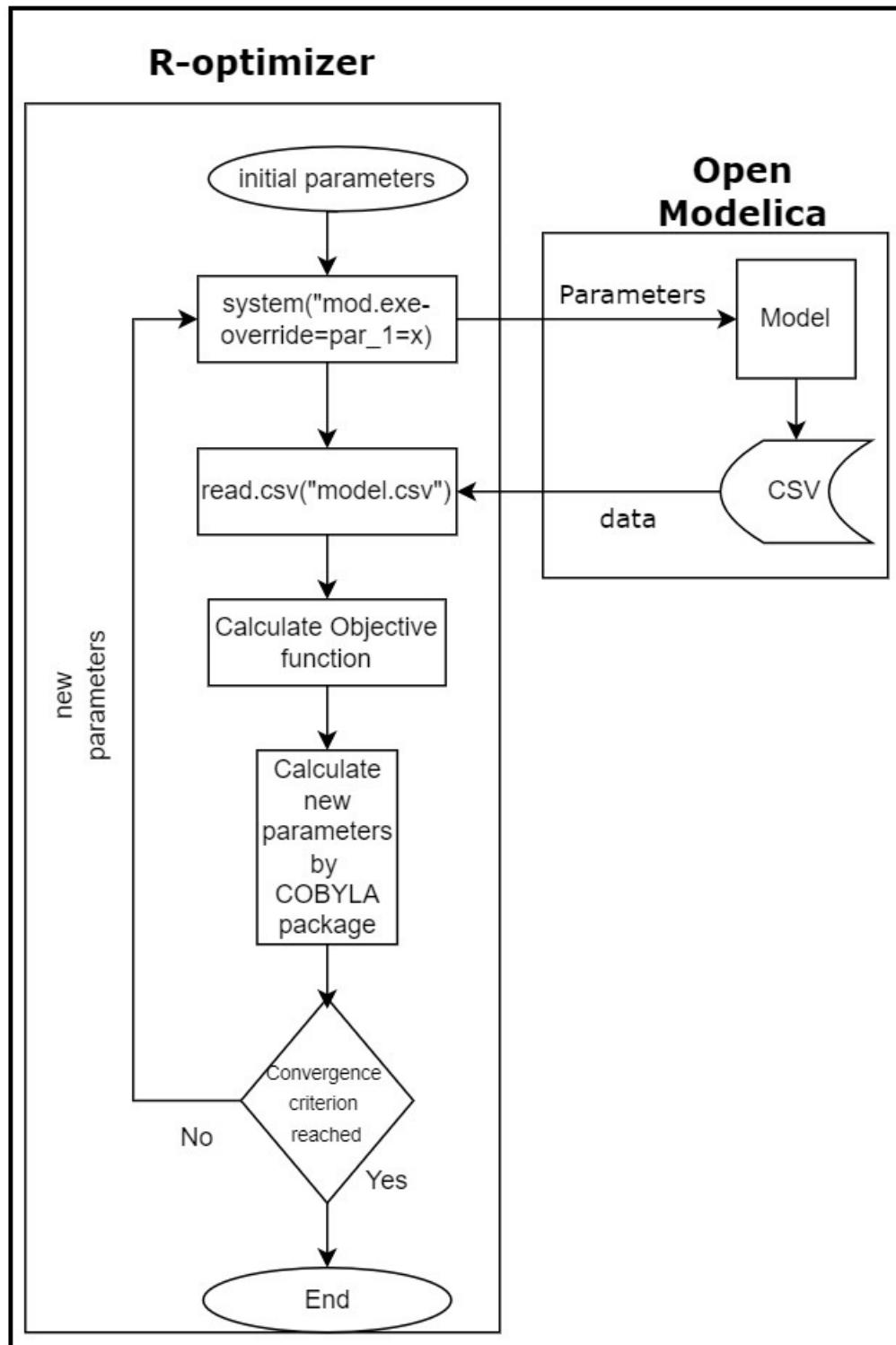


Figure 4.3: Figure showing the interoperability of R-lang and OpenModelica

This page was intentionally left blank.

Chapter 5

Optimization of vapor phase model

There is a question of which objective function to be used in the optimization, Absolute Error or Mean Square Error. Absolute Error is defined as:

$$F \rightarrow \frac{1}{n} \frac{\sum_{i=1}^{17} |x_{i,end}^{plant} - x_{i,end}^{model}|}{x_{i,end}^{plant}} \rightarrow \min \quad (5.1)$$

Mean Square Error is defined as:

$$F \rightarrow \frac{1}{n} \sqrt{\sum_{i=1}^{17} \left(\frac{x_{i,end}^{plant} - x_{i,end}^{model}}{x_{i,end}^{plant}} \right)^2} \rightarrow \min \quad (5.2)$$

The objective functions in this simulation have constraints and these are defined as:

$$\text{subject to } F(E_i, K0_i) = 0 \quad (5.3)$$

$$E_j \geq 0 \quad (5.4)$$

$$-E_j \geq -400 \quad (5.5)$$

$$K0_j \geq 0 \quad (5.6)$$

$$-K0_j \geq -1e+50 \quad (5.7)$$

Where $x_{i,end}^{plant}$ represents the mass fraction of component i at the end of the reactor or the outlet of the reactor from plant data. $x_{i,end}^{model}$ is the endpoint composition of component i from the model. n represents the number of components, here it is 17. Tuning is done only for the activation energies, but not for the pre-exponential factor. We chose only the activation energies to tune as they are more sensitive to tune. We allowed activation energies to change $\pm 50\%$ of the values given in the Enikeeva et al. (2021). Also, the profiles given in the Enikeeva et al. (2021) were used as the plant data in this study.

5.1 Absolute Mean Error

We chose the termination criterion of the optimization problem as the difference in successive values of parameters to be less than $1e-6$. The starting parameters are mentioned in the Table 3.3. It took 1530 function evaluations to converge. The results are shown in the figures 5.1, 5.2, 5.3, 5.4 and 5.5.

Although temperature is not considered in the objective function, the fact that the temperature after cooler was modeled shows that the temperature effects are already included. The Enikeeva et al. (2021) has 3 reactors with a inter-cooler after the second reactor. Because of this and the temperature at the entrance is being identical in the Enikeeva et al. (2021), it shows that the temperature effects are already considered in the objective function. This approach obviates the need to assign a suitable weighting factor for the temperature error in the objective function. For example the weighting factor in the temperature error should be of the order of 0.01 or 0.001.

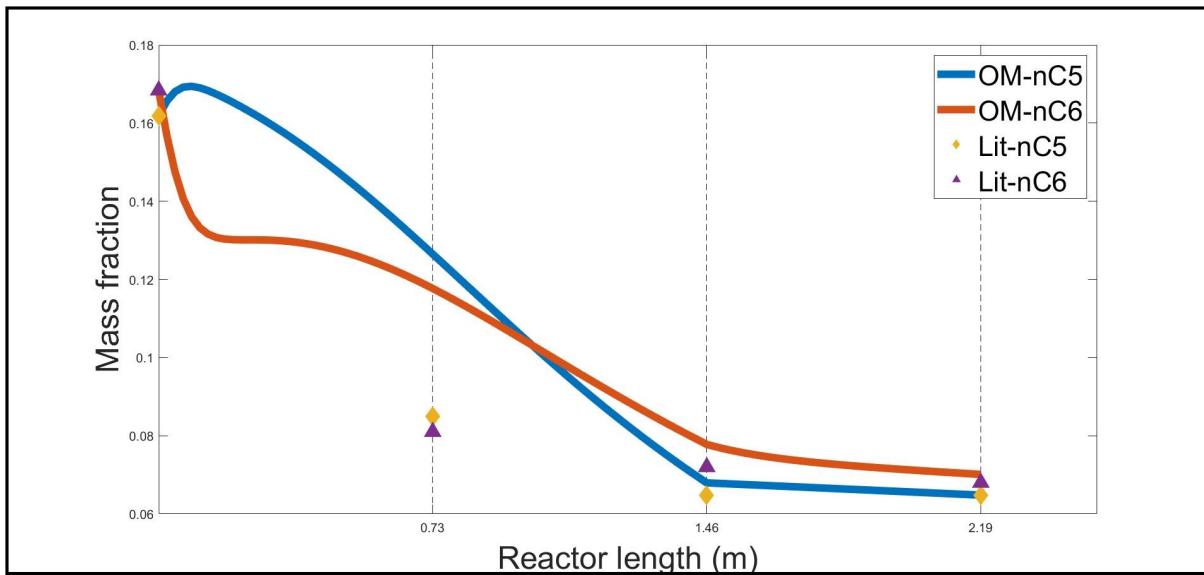


Figure 5.1: Figure showing the concentration profile of n-paraffins across the reactor length after tuning with Absolute Error as objective function. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.

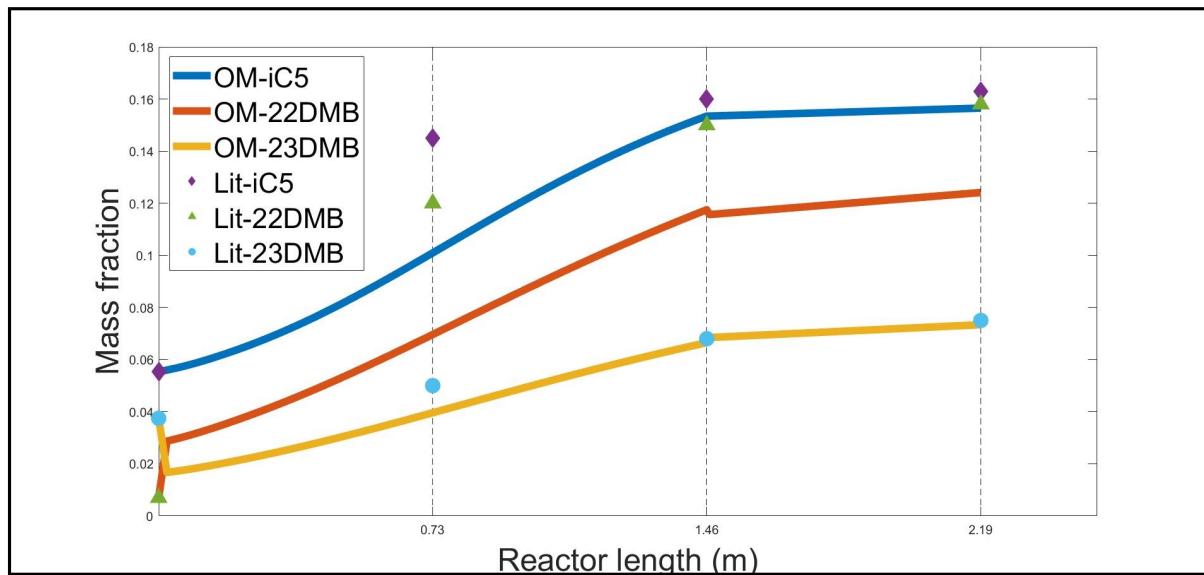


Figure 5.2: Figure showing the concentration profile of value added components across the reactor length after tuning with Absolute Error as objective function. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.

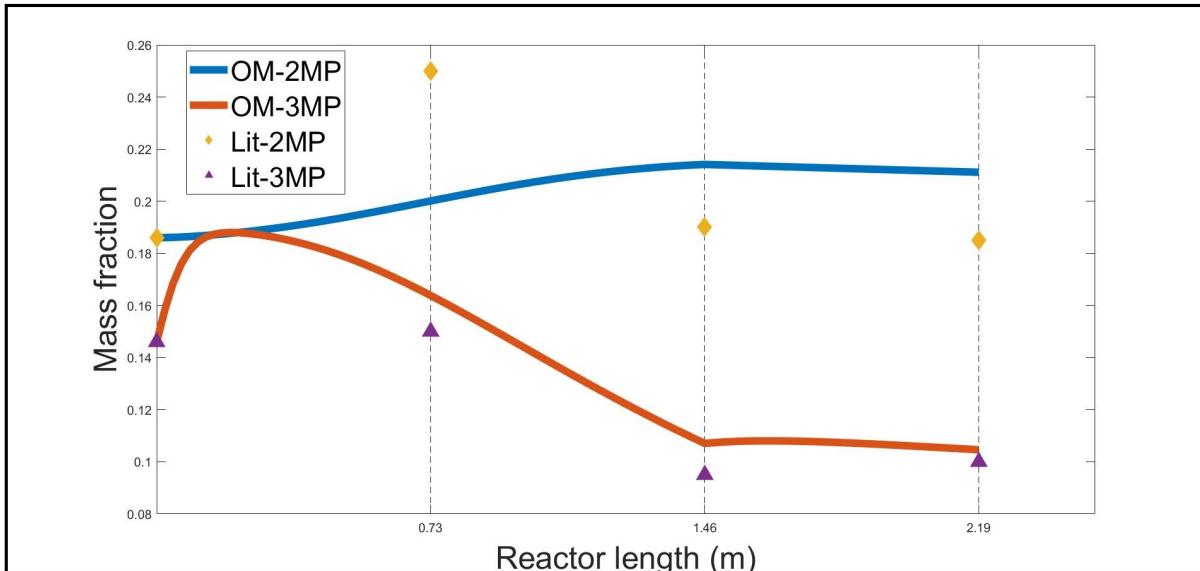


Figure 5.3: Figure showing the concentration profile of 2MP and 3MP across the reactor length after tuning with Absolute Error as objective function. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.

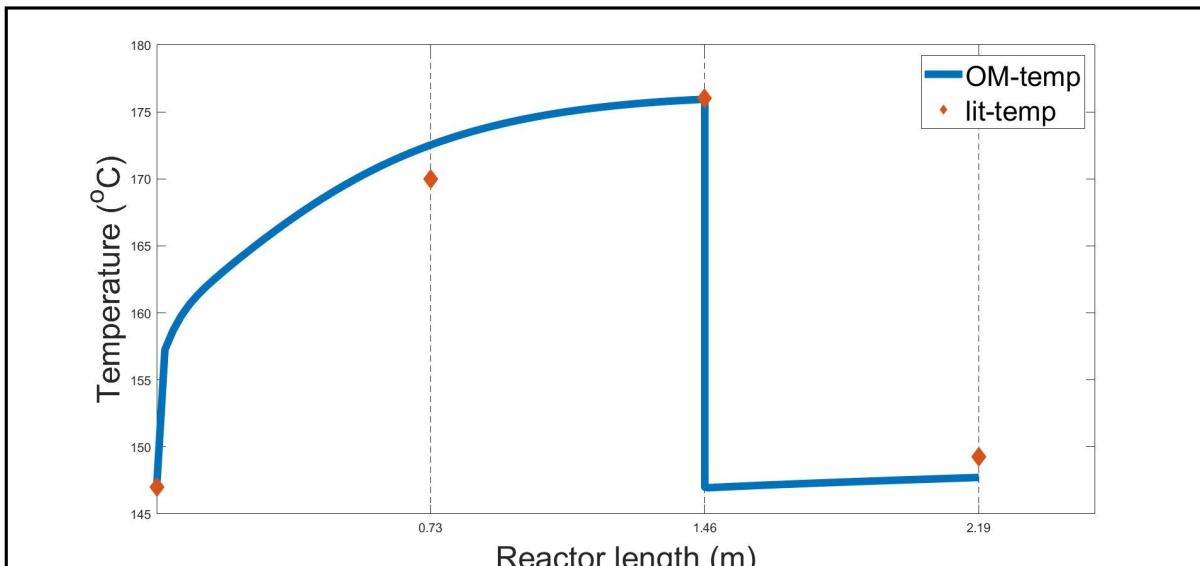


Figure 5.4: Figure showing temperature profile across the reactor length after tuning with Absolute Error as objective function. The sudden drop in the temperature profile at 1.46m is due to the cooler. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.

5.2 Mean Square Error (MSE)

In this section we look at the performance of Mean Square Error based objective function. An attempt was made to tune the parameters with objective function as equation 5.2. Convergence



Figure 5.5: Figure showing the comparison of outlet composition(weight fraction) from both Enikeeva et al. (2021) and Model after tuning with Absolute Error as objective function. The optimization objective function contains the end point data only.

Table 5.1: Table showing the performance parameters for Absolute Error as objective function.

S.No.	Performance Parameter	Model Prediction	literature (Enikeeva et al., 2021)	%Error
1	iC5/C5P	70.68%	71.55%	1.22%
2	22DMB/C6P	21.23%	26.96%	21.25%
3	23DMB/C6P	12.55%	12.80%	1.92%
4	(2MP+3MP)C6P	54.17%	48.63%	11.39%
5	PIN	1.04	1.11	6.15%
6	RON	77.68087	79.130	0.53%

criterion is same as before namely successive parameter values in the iteration must be less than 1e-6. As the simulations didn't converge within 1530 function evaluations we terminated the optimization. Recall that the 1530 is the number of evaluations required in the previous section. The corresponding profiles are given in Figures 5.6, 5.7, 5.8, 5.9 and 5.10. One can

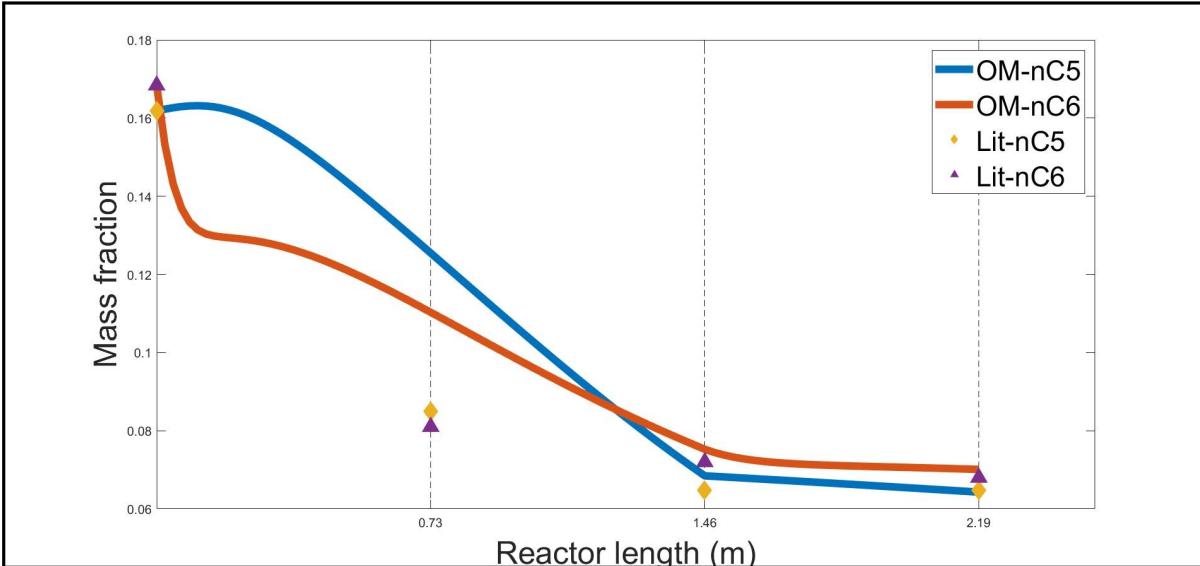


Figure 5.6: Figure showing the concentration profile of n-paraffins across the reactor length for the end point tuning. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.

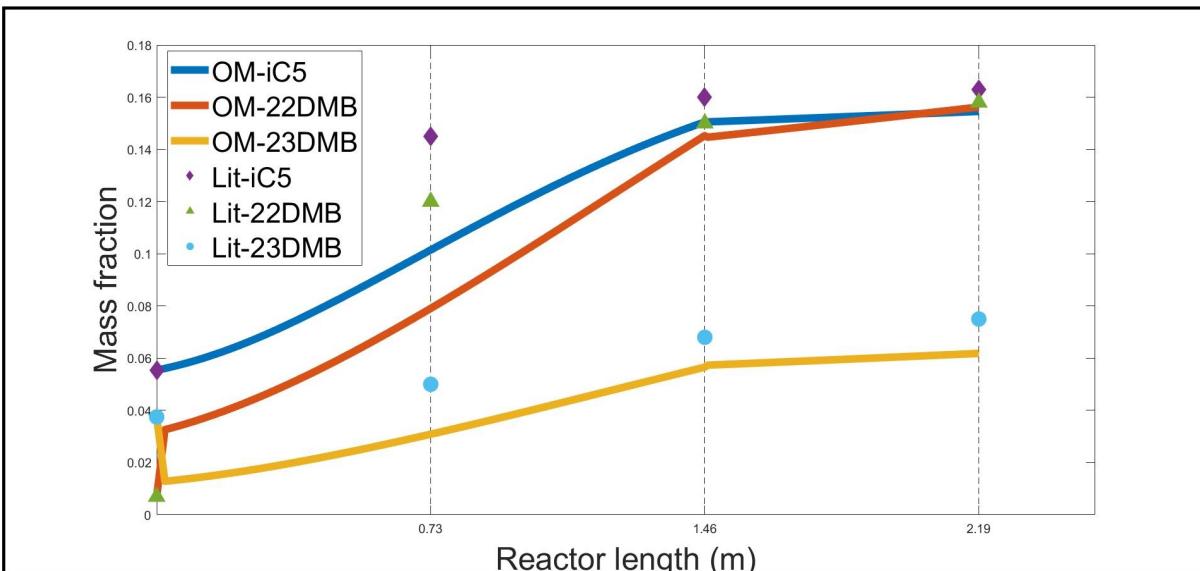


Figure 5.7: Figure showing the concentration profile of value added components across the reactor length after tuning with Mean Square Error as objective function. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.

observe from the errors in Table 5.1 and Table 5.2, and the corresponding figures, the errors in the parameters are less for MSE when compared to Absolute Error for the same number of function evaluations. As a result, we will work only with MSE from now on.

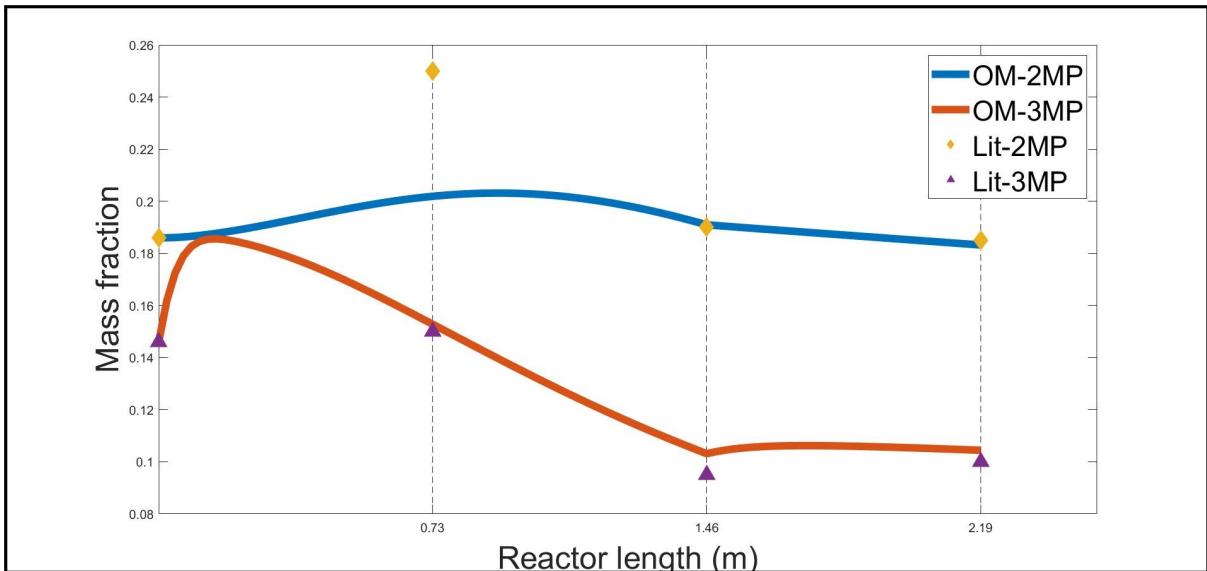


Figure 5.8: Figure showing the concentration profile of 2MP and 3MP across the reactor length after tuning with Mean Square Error as objective function. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.

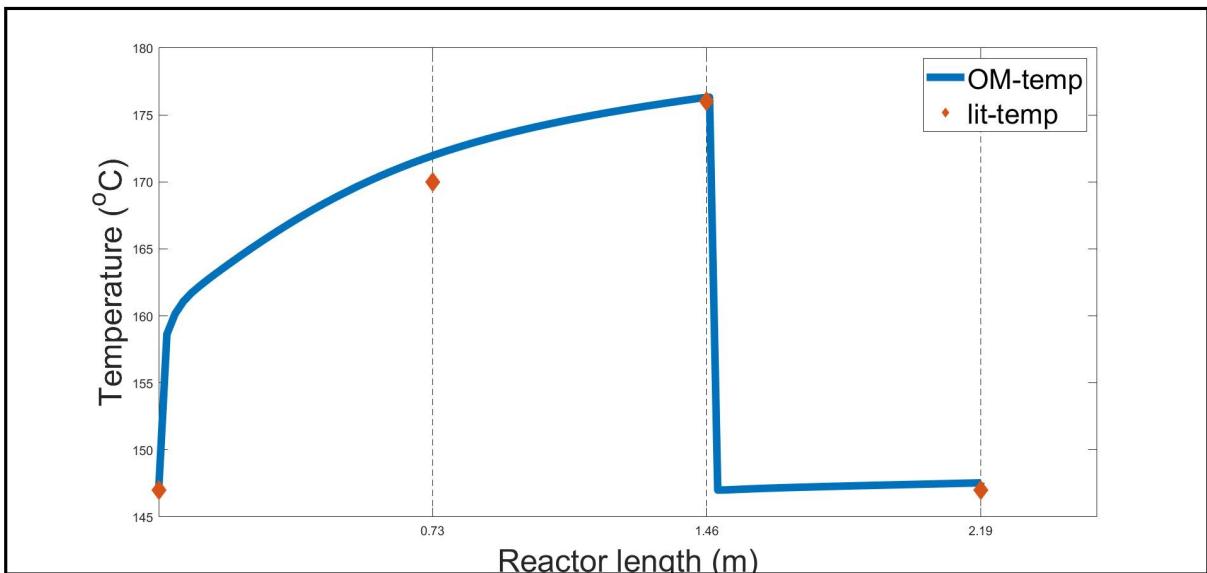


Figure 5.9: Figure showing temperature profile across the reactor length after tuning with Mean Square Error as objective function. The sudden drop in the temperature profile at 1.46m is due to the cooler. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.

5.3 Reducing the Number of Parameters

One criticism about this work is the large number of parameters used in tuning. An attempt is made in this section to reduce the number of parameters.

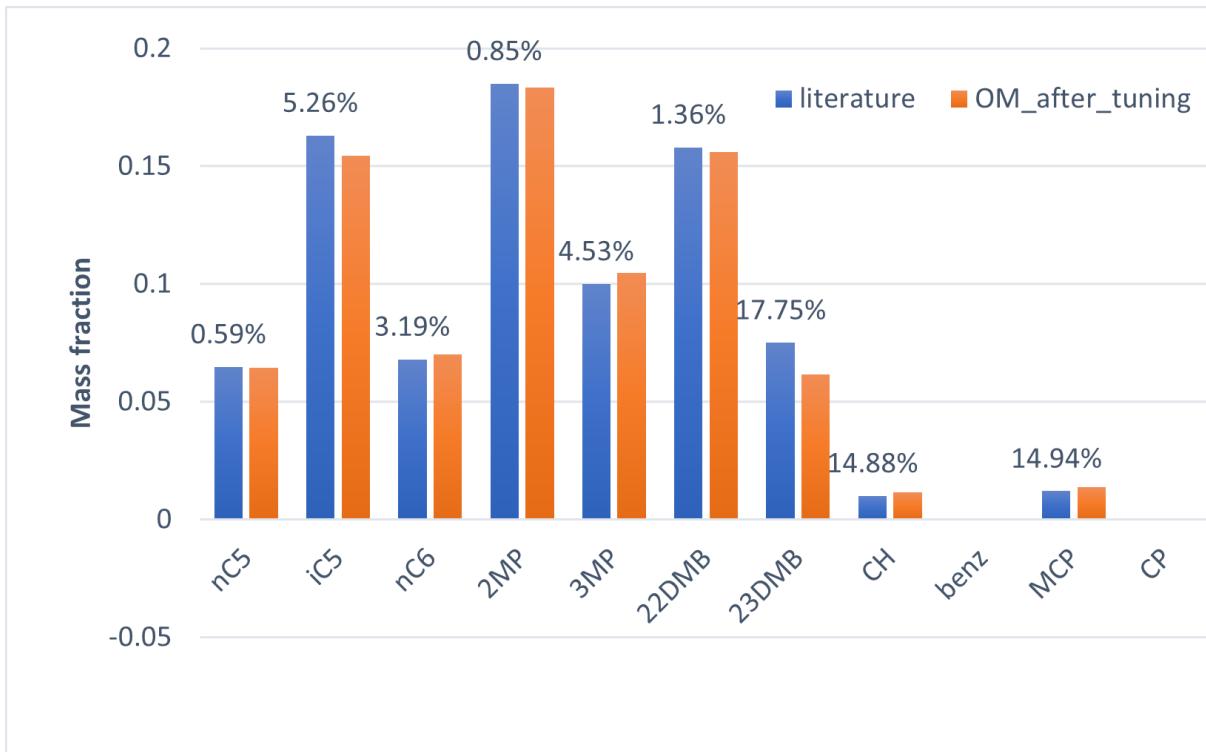


Figure 5.10: Figure showing the comparison of outlet composition(weight fraction) from both Enikeeva et al. (2021) and Model after tuning with Mean Square Error as objective function. The optimization objective function contains the end point data only.

Table 5.2: Table showing the performance parameters for MSE as objective function.

S.No.	Performance Parameter	Model Prediction	Literature (Enikeeva et al., 2021)	%Error
1	iC5/C5P	70.56%	71.55%	1.384%
2	22DMB/C6P	27.07%	26.96%	0.409%
3	23DMB/C6P	10.72%	12.80%	16.270%
4	(2MP+3MP)/C6P	50.02%	48.63%	2.853%
5	PIN	1.08	1.1135	2.662%
6	RON	78.063	79.130	0.038%

The values in Table 5.3 shows that there are 15 parameters which have deviated a lot. But, the first 38 reactions are more important than the rest of the reactions, because the remaining are hydrocracking reactions, which are insignificant. Hence, there are now 13 parameters that are more important. In this section we are going to tune only these 13 parameters and compare the results. The results are shown in Figures 5.11, 5.12, 5.13, 5.14 5.15.

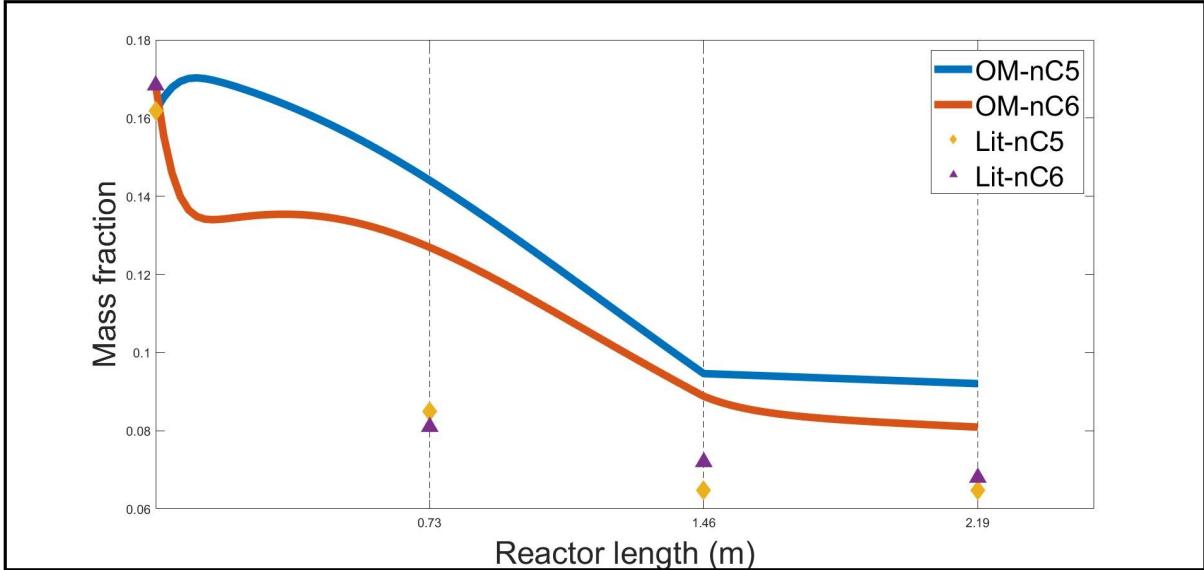


Figure 5.11: Figure showing the concentration profile of n-paraffins across the reactor length after tuning with selected parameters. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.

One can see that the simulated profiles are less accurate than the case with all parameters tuned. The situation does not improve even when the tolerance is reduced to 1e-9. These experiments show that the way we shortlisted the parameters to tune is incorrect. Some detailed sensitivity studies have to be undertaken to decide whether we can achieve good results by tuning less number of parameters. We may also have to choose a few pre-exponential factors for better accuracy. As the magnitudes of these parameters are vastly different, a lot of care may have to be exercised in choosing the weighting factors.

The chosen method of using OM for simulation is a lot more efficient compared to Aspen-Hysys, which takes about 10 minutes to simulate the reactor using the module PFR. In comparison, OM requires only a few seconds to do the simulation. In this work, we have carried out several optimization studies with a large number of function evaluations, of the order of 1,500 and more. Each function evaluation involves one simulation. If each function evaluation requires 10 minutes, it would have taken several days to do one optimization study in Aspen-Hysys. As OM is more than 100 times faster, it has been possible to carry out the optimization studies in a reasonable time.

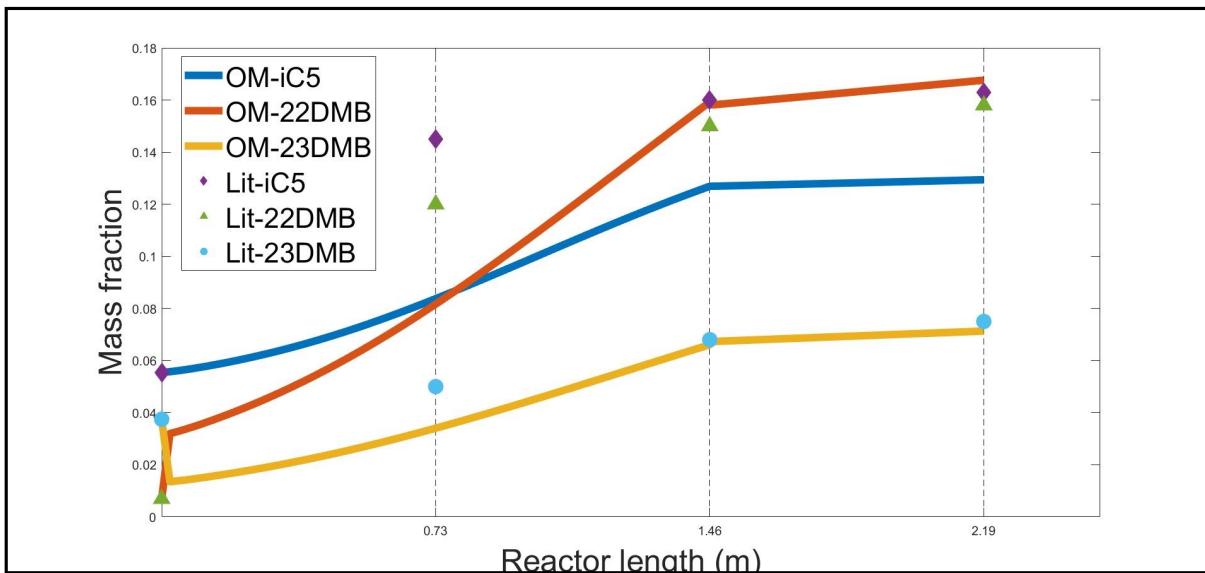


Figure 5.12: Figure showing the concentration profile of value added components across the reactor length after tuning with selected parameters. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.

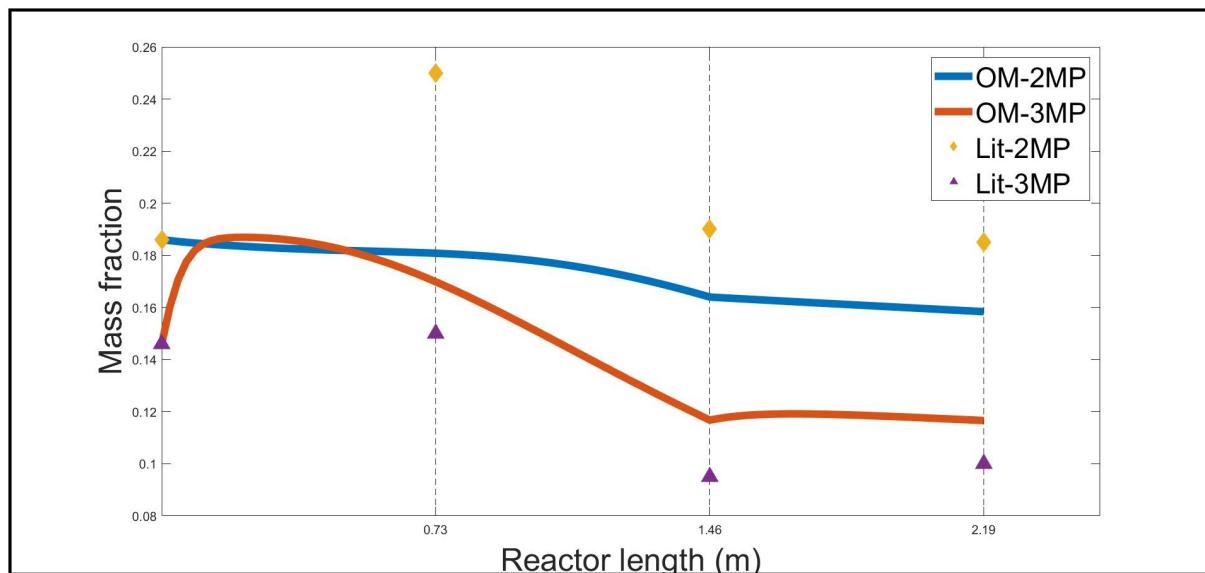


Figure 5.13: Figure showing the concentration profile of 2MP and 3MP across the reactor length after tuning with selected parameters. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.

Table 5.3: Table showing the converged values of activation energies and their deviation from the original parameters given in Table 3.3 on Page 26 (Enikeeva et al., 2021). MSE is used for tuning.

rxn	Ej (kmol/hr) Tuned Parameters	%Deviation	rxn	Ej (kmol/hr) Tuned Parameters	%Deviation
1	146.2486	1.80%	28	179.5446	38.87%
2	197.343	27.91%	29	139.3904	9.80%
3	191.3387	33.64%	30	97.44449	1.20%
4	150.819	0.39%	31	139.1547	7.41%
5	150.4666	0.34%	32	93.36829	8.78%
6	155.5116	0.26%	33	166.82	0.78%
7	152.3545	0.40%	34	80.71469	11.01%
8	147.8399	1.41%	35	182.1682	2.73%
9	111.6752	12.26%	36	234.3015	5.12%
10	183.2534	31.77%	37	59.72879	0.12%
11	64.79836	0.46%	38	53.97367	0.20%
12	78.10174	1.35%	39	325.2708	1.52%
13	139.9071	4.27%	40	310.5722	5.62%
14	220.5254	37.59%	41	282.9822	0.70%
15	99.21494	0.95%	42	165.9584	0.22%
16	120.3178	14.15%	43	164.2713	0.93%
17	3.336818	4.93%	44	109.9923	1.84%
18	4.366529	8.84%	45	258.3823	2.50%
19	224.1559	24.39%	46	267.163	1.20%
20	486.1274	21.40%	47	343.3427	30.30%
21	187.0113	0.02%	48	258.7136	2.37%
22	300.2305	0.19%	49	326.4193	10.42%
23	47.10999	7.77%	50	299.2719	1.45%
24	353.7568	3.47%	51	278.5655	5.63%
25	109.8432	15.34%	52	308.7079	5.00%
26	117.7835	32.88%	53	276.6895	5.96%
27	112.8877	16.66%	54	353.514	26.79%

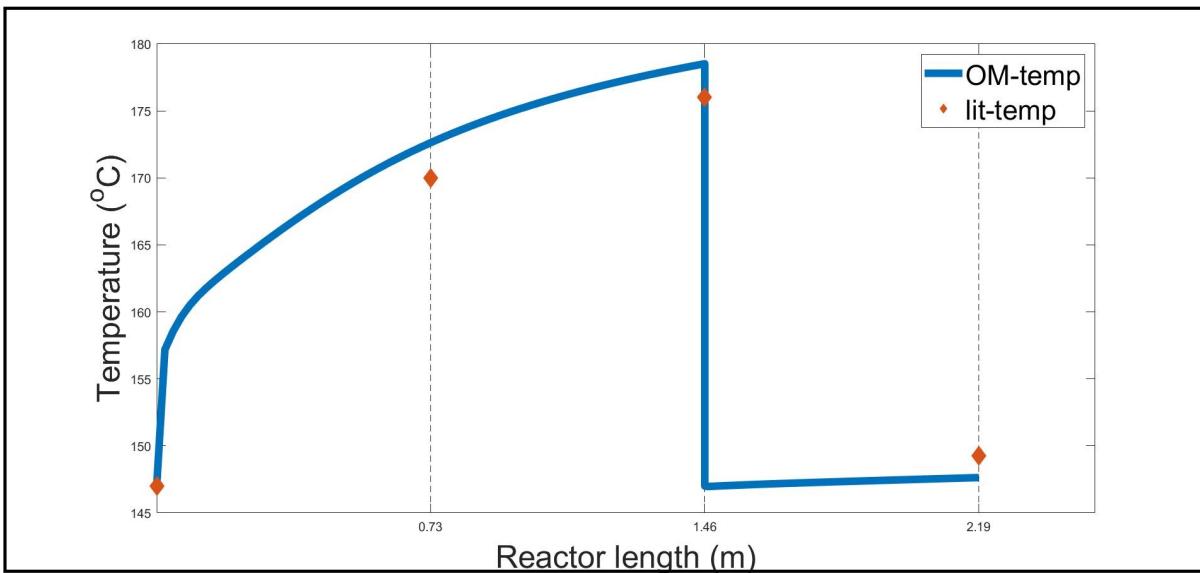


Figure 5.14: Figure showing temperature profile across the reactor length after tuning with selected parameters. The sudden drop in the temperature profile at 1.46m is due to the cooler. The solid lines represent the simulation results and the points represent the data from Enikeeva et al. (2021). Vertical grid lines showing the reactor end.

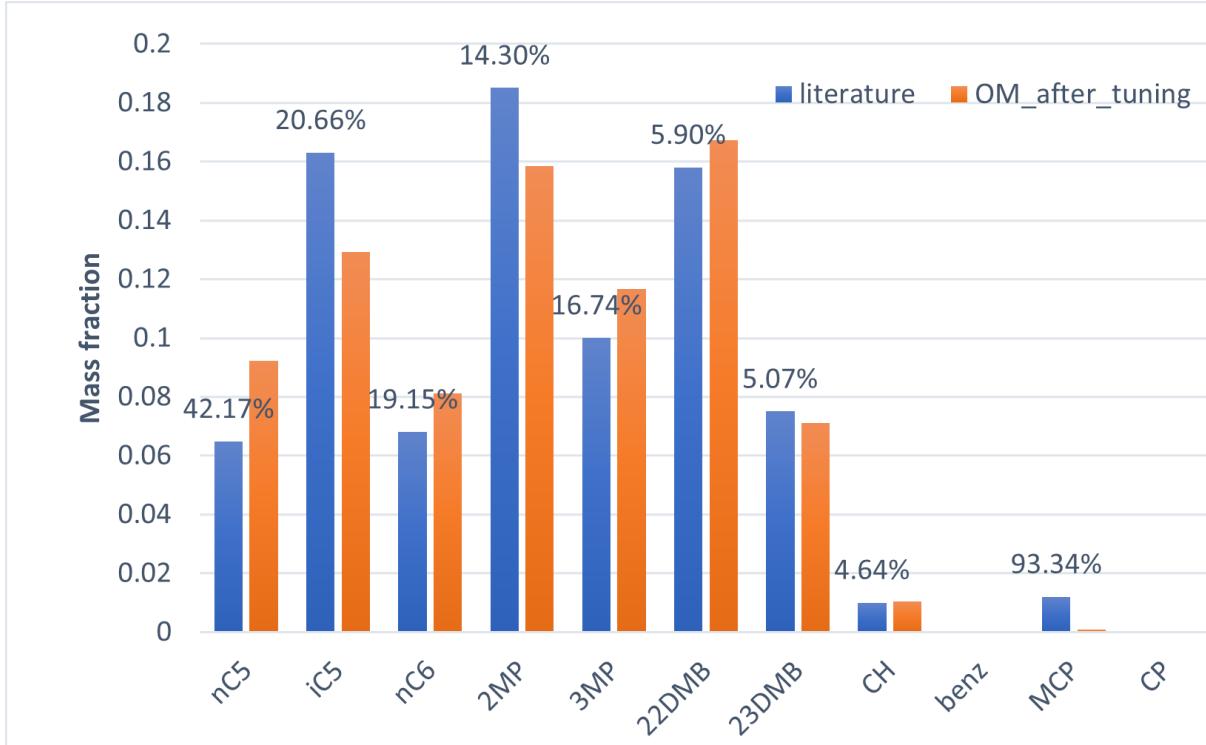


Figure 5.15: Figure showing the comparison of outlet composition(weight fraction) from both Enikeeva et al. (2021) and Model after tuning with selected parameters.

Chapter 6

Liquid Phase Reactor Simulation

In the previous chapter, we considered the case of reactions taking place in the Vapor phase. This is because of Hydrogen being present in large quantities, see Table 3.4 on Page 27. It is of the order of 80%. Some reactors, on the other hand, operate at a much lower Hydrogen concentration, less than 30%. These reactants will be in mixed phase, consisting of both liquid and vapor. If the liquid phase is substantial, reaction in this phase will dominate in the sense that the vapor may not be able to penetrate the catalyst particles and react. Only liquid phase reaction will take place in this case. In this chapter, this situation is considered by simulating a solved example from Aspen-Hysys, with input values as in Table 6.3.

The kinetic constants will be different for liquid phase reaction from the vapor phase reaction. In particular, the pre-exponential factor and the activation energy of the rate constant could change. Although we may not know these values, these can be estimated during the tuning process. Only the activation energy has been tuned earlier, as it is a more sensitive parameter compared to the pre-exponential factor. If solubility is an important factor, tuning of the pre-exponential factor will take care of this effect. As the plant data are not available, however, only the activation energy is tuned in this chapter as well.

As mentioned above, the input conditions as given in Table 6.3 will be considered in this

Table 6.1: Table showing the design parameters of the ISOM reactor (Aspen-Hysys)

S.No.	Design Parameter	Value
1	Density of catalyst (kg/m^3)	2500
2	Void fraction (ϕ)	0.5
3	Length of the reactor (m)	7.69*2
4	Diameter of the reactor (m)	2.0

Table 6.2: Table showing the operating conditions of the stream entering the reactor (Aspen-Hysys)

S.No.	Design Parameter	Value
1	Flow rate ($kmol/hr$)	550
2	Temperature (oC)	162
3	Pressure (kPa)	2550

chapter. The corresponding operating conditions are given in Table 6.2. The original Aspen-Hysys example has nC_7H_{16} , which is not considered in the model of this work. As a result, this component is neglected, and the remaining compositions normalized. Parameters in section 3.3 are used in the current simulation.

The results are shown in Figures 6.1, 6.2, 6.3 and 6.4. As vapor phase kinetic parameters have been used in this simulation, the profiles are not necessarily correct, and the end points differ from values given in Aspen-Hysys. Using the method described in the previous chapter, one can tune the parameters if plant data are available. We do not have access to any plant data at the time of writing this report.

6.1 Optimization of Liquid Phase Model

In this section optimization upon the Liquid phase reaction model is being discussed. We are going to take the Mean Square Error as the objective function. It was mentioned previously that this function performs better. The starting activation energy parameters to run the optimization is taken from Table 3.3. A limit of $\pm 15\%$ is considered for the activation energy. In this method

Table 6.3: Table showing the composition of the stream entering the ISOM reactor (Aspen-Hysys)

S.No.	Component	Molar flow	Mole fraction
1	nC5	108.1266744	0.194615197
2	iC5	59.55329663	0.107188875
3	nC6	82.90315179	0.149215846
4	2MP	39.66378797	0.071390117
5	3MP	26.74985226	0.048146563
6	22DMB	2.539670266	0.004571105
7	23DMB	5.165388492	0.009297087
8	CH	14.15	0.025468323
9	benzene	8.279327144	0.014901807
10	H2	155.1984875	0.27933888
11	MCP	35.9507588	0.064707104
12	CP	11.79028372	0.021221113
13	nC4	1.975246118	0.003555209
14	iC4	0.348156285	0.00062664
15	nC3	0.934851154	0.001682621
16	C2	1.169050036	0.002104151
17	C1	1.094161484	0.001969361

to prevent from reaching a local minima the optimization problem is solved in parts. In the first run a limit of 1000 function evaluations is set and this run gives some new parameters. The values obtained is taken as the starting point for the next 1000 iterations. The results are shown in Figures 6.5, 6.6, 6.6 and 6.8 and 6.9. Performance parameters are shown in Table 6.4. The errors in performance parameters are not more than 10% which shows that our ISOM model integrated with optimizer is working good.

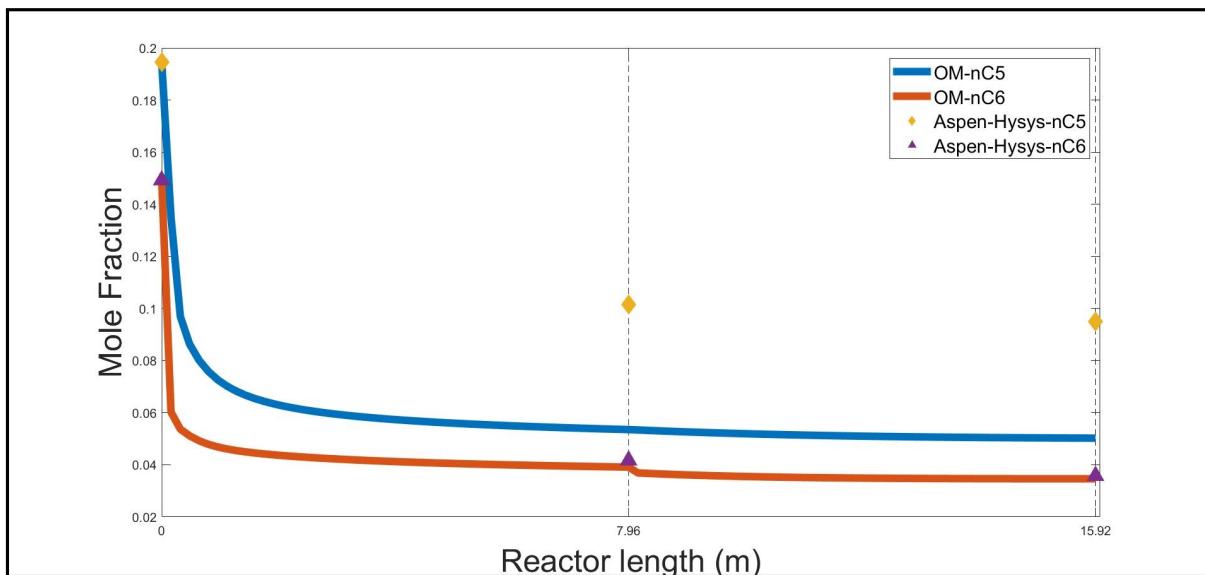


Figure 6.1: comparison plots of n-paraffins between Aspen-Hysys and OM model. Here, vertical bar represents the end of the reactor. Solid lines represent OM results and the dots represents Aspen-Hysys data.

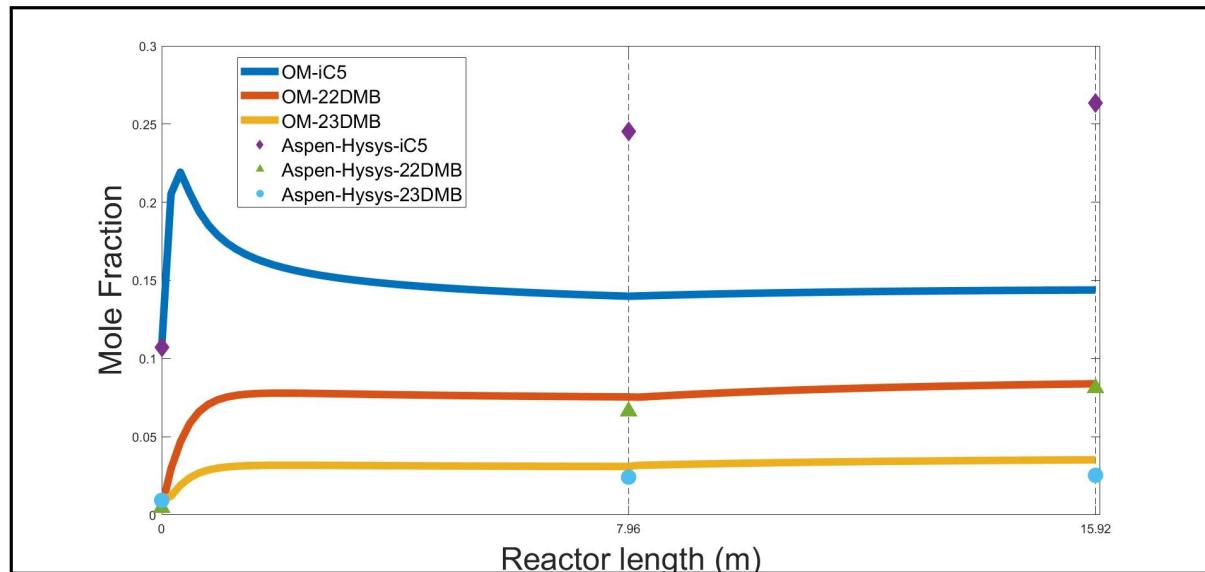


Figure 6.2: Comparison plots of value added components obtained between Aspen-Hysys and OM model. Here, vertical bar represents the end of the reactor. Solid lines represent OM results and the dots represents Aspen-Hysys data.

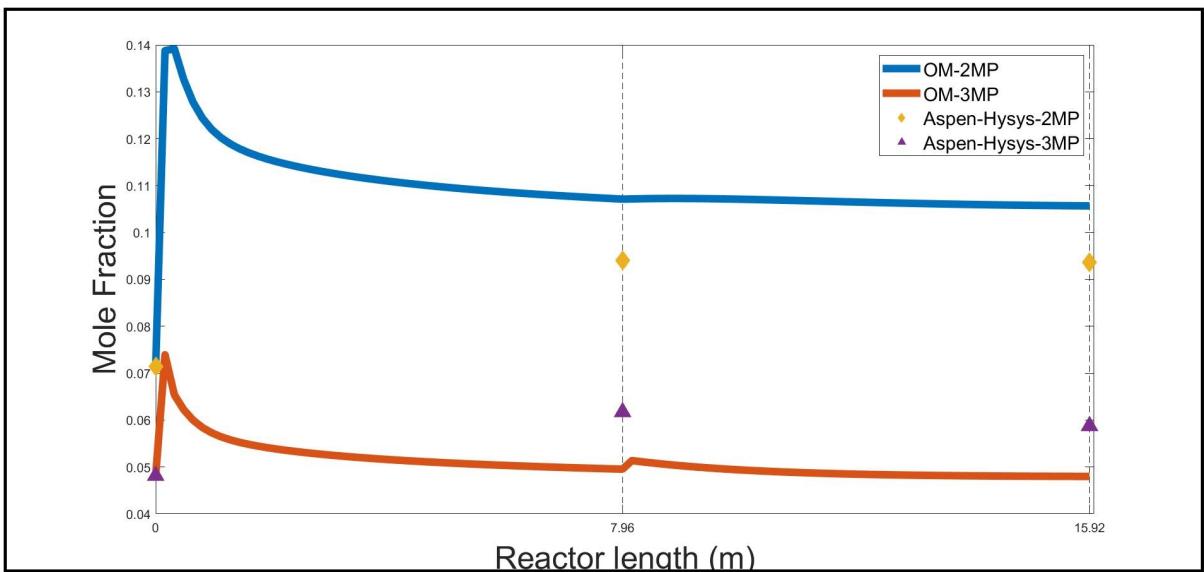


Figure 6.3: Comparison plots of 2MP and 3Mp between Aspen-Hysys and OM model. Here, vertical bar represents the end of the reactor. Solid lines represent OM results and the dots represents Aspen-Hysys data.

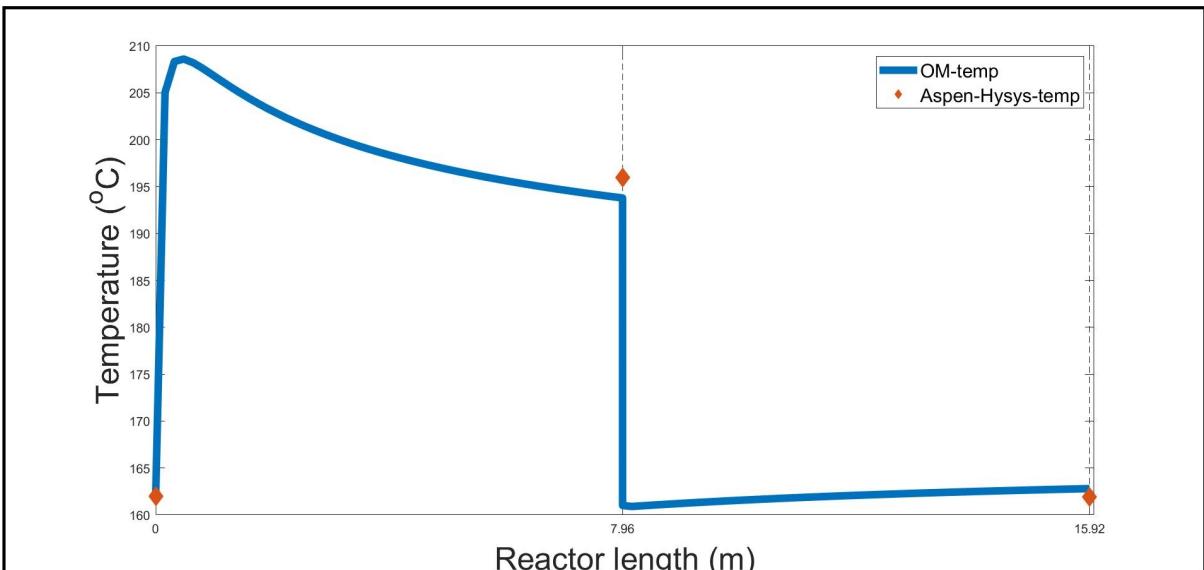


Figure 6.4: Comparison plots of Temperature profile between Aspen-Hysys and OM model. Here, vertical bar represents the end of the reactor. Solid lines represent OM results and the dots represents Aspen-Hysys data.

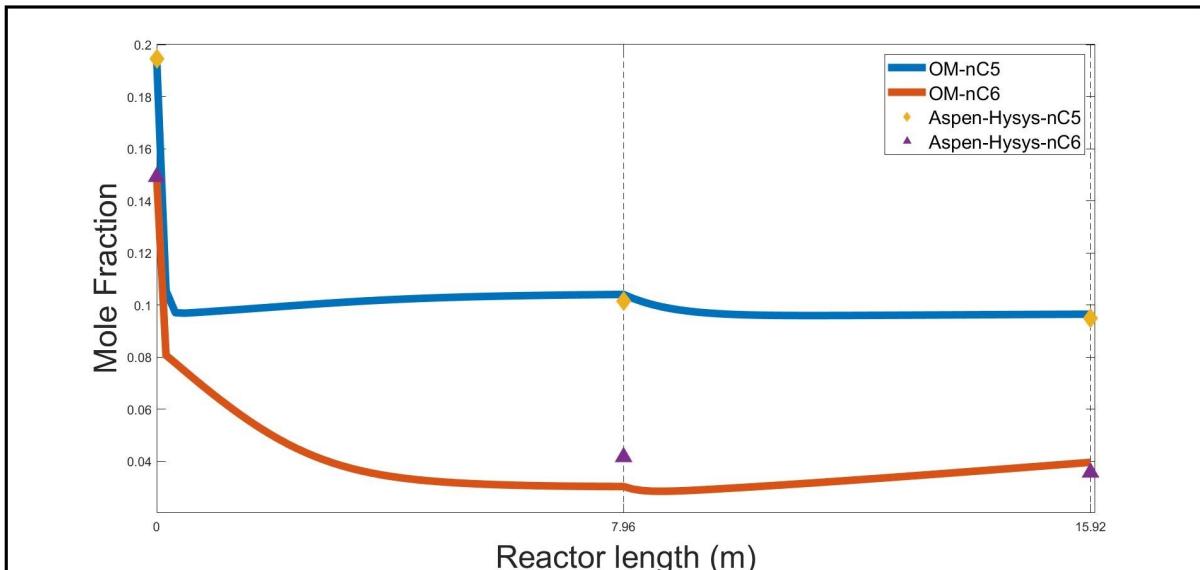


Figure 6.5: Figure showing the concentration profile of n-paraffins across the reactor length after tuning. The solid lines represent the simulation results and the points represent the data from Aspen-Hysys. Vertical grid lines showing the reactor end.

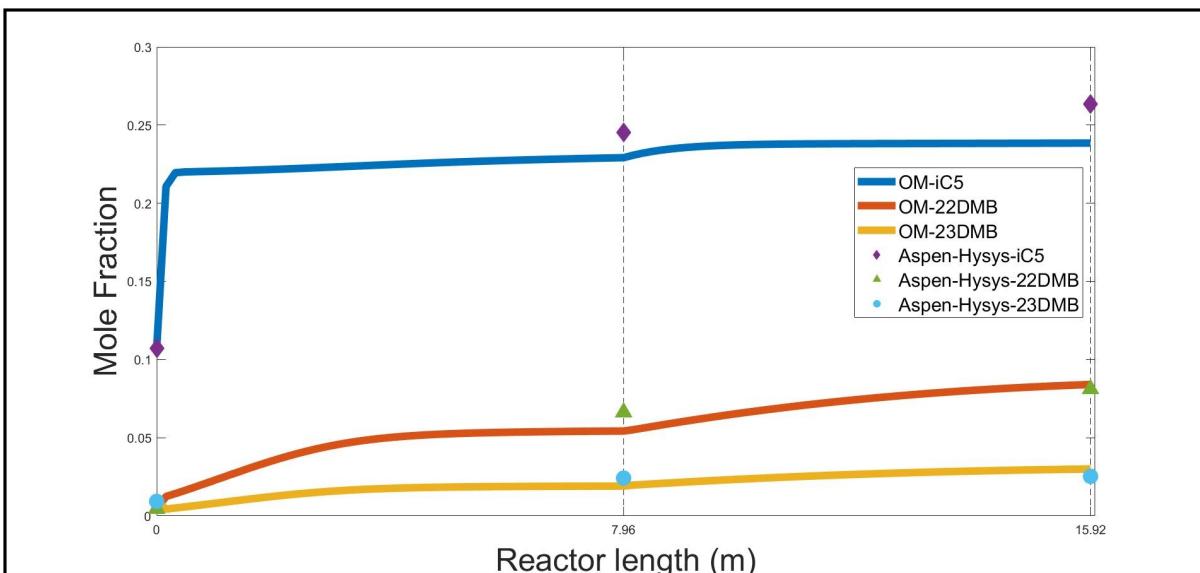


Figure 6.6: Figure showing the concentration profile of value added components across the reactor length after tuning. The solid lines represent the simulation results and the points represent the data from the Aspen-Hysys Simulation. Vertical grid lines showing the reactor end.

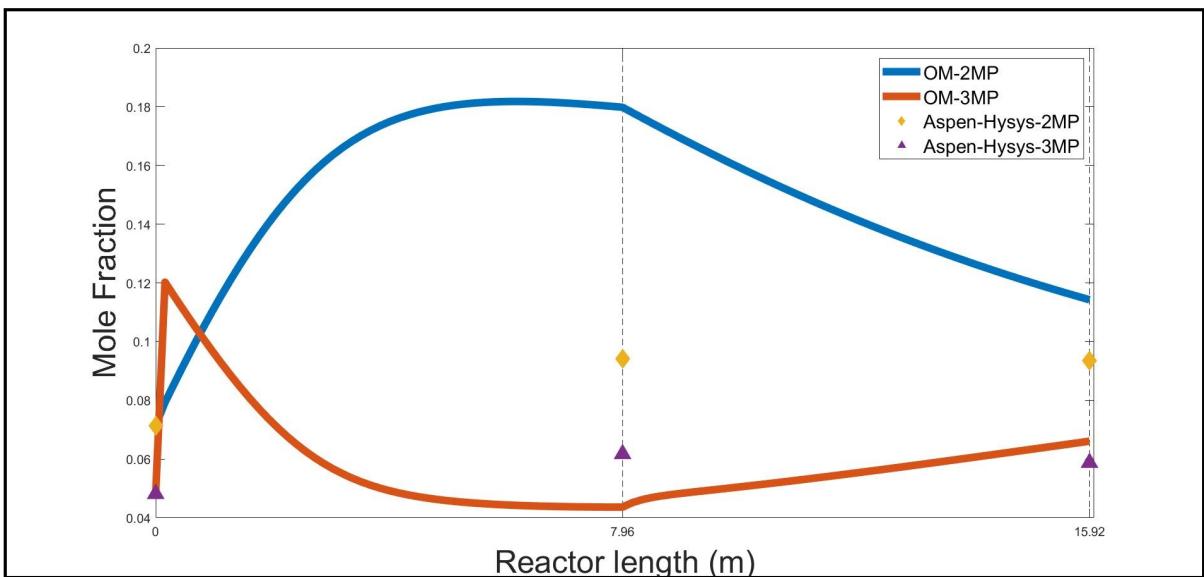


Figure 6.7: Figure showing the concentration profile of 2MP and 3MP across the reactor length after tuning. The solid lines represent the simulation results and the points represent the data from the Aspen-Hysys Simulation. Vertical grid lines showing the reactor end.

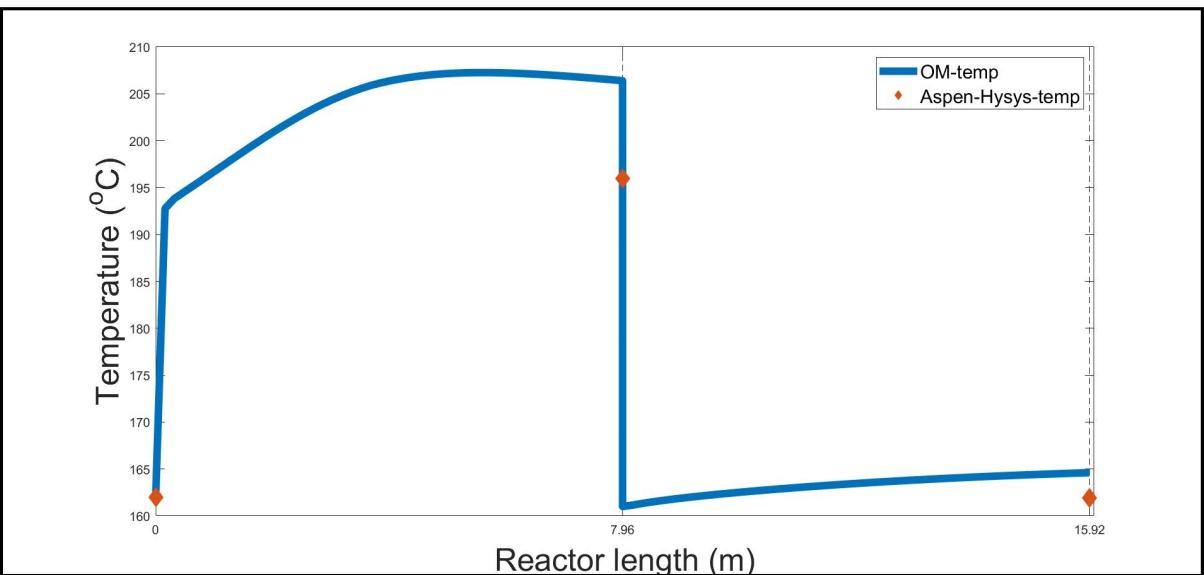


Figure 6.8: Figure showing temperature profile across the reactor length after tuning. The sudden drop in the temperature profile at 1.46m is due to the cooler. The solid lines represent the simulation results and the points represent the data from Aspen-Hysys Simulation. Vertical grid lines showing the reactor end.

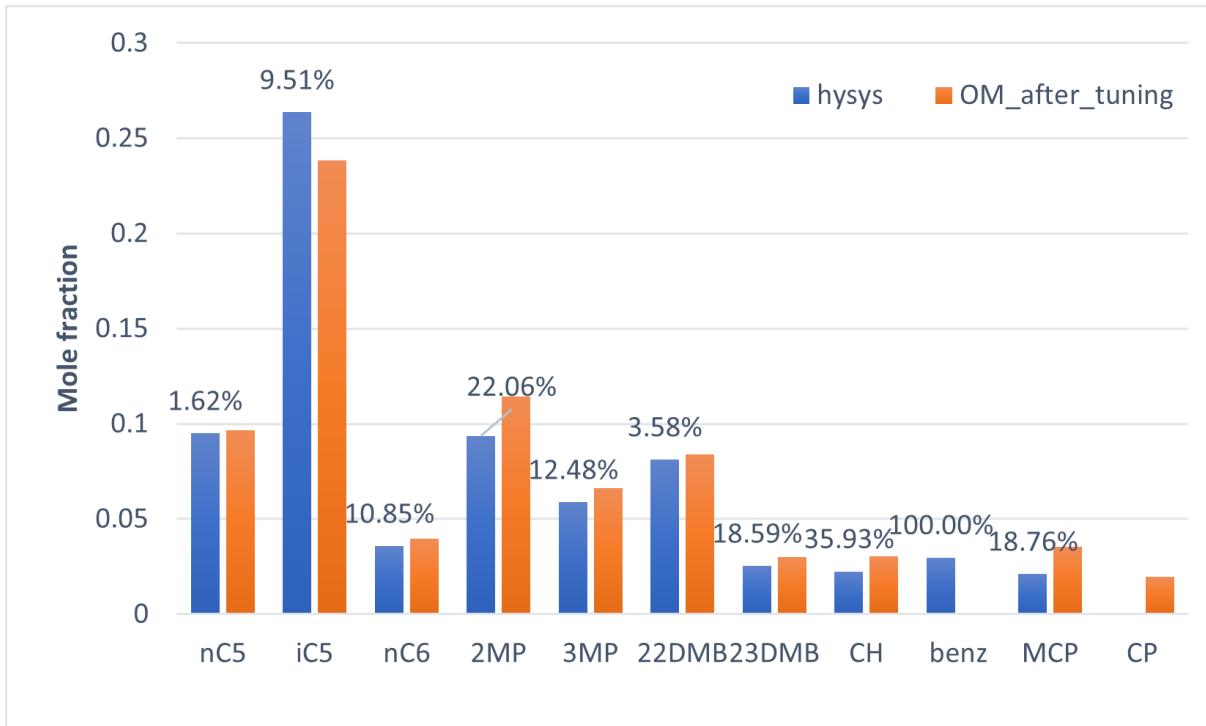


Figure 6.9: Figure showing the comparison of outlet composition(weight fraction) from both Aspen-Hysys Simulation and OM Model after tuning.

Table 6.4: Table showing the comparison of performance parameters between Aspen-Hysys Simulation and OM Model after tuning.

S.No.	Performance Parameter	Aspen Hysys	Model Prediction	%Error
1	iC5/C5P	73.51%	71.19%	3.154%
2	22DMB/C6P	31.36%	28.55%	8.945%
3	23DMB/C6P	9.78%	10.19%	4.257%
4	(2MP+3MP)C6P	58.87%	61.26%	4.0589%
5	PIN	1.15	1.10	4.106%
6	RON	80.76	79.91	1.052%

Chapter 7

Conclusion and Future work

A rigorous model that describes the reactions taking place in an isomerization reactor is developed in OpenModelica. An optimization method to tune the kinetic parameters so as to match plant data is also developed using a function based optimizer COBYLA, available in the statistical software R. A software interface was developed to make R communicate with OpenModelica. Literature values (Enikeeva et al., 2021) have been taken as plant data to tune the model parameters. Mean Square Error criterion has been found to be more accurate than absolute error criterion in optimization. Both liquid phase and vapor phase reactions have been considered in this work.

Many isomerization reactors in India seem to operate in liquid phase. The proposed model will be validated with plant data, when available. Although an attempt has been made in this work to reduce the number of tuning parameters, the results are not satisfactory. This important work is left to a future study. Integration of the isomerization reactor with a crude distillation column is left for future work.

This page was intentionally left blank.

References

- Ahmed, A. M., Jarullah, A. T., Abed, F. M., Mujtaba, I. M., 2018. Modeling of an industrial naphtha isomerization reactor and development and assessment of a new isomerization process. *Chemical Engineering Research and Design* 137, 33–46.
URL <https://www.sciencedirect.com/science/article/pii/S0263876218303289>
- Awan, Z., Kazmi, B., Hashmi, S., Raza, F., Hasan, S., Yasmin, F., 2021. Process system engineering (pse) analysis on process and optimization of the isomerization process. *Iranian Journal of Chemistry and Chemical Engineering (IJCCE)* 40 (1), 289–302.
URL http://www.ijcce.ac.ir/article_37243.html
- Chekantsev, N. V., Gyngazova, M. S., Ivanchina, E. D., 2014. Mathematical modeling of light naphtha (c5, c6) isomerization process. *Chemical Engineering Journal* 238, 120–128, proceedings of the XX International conference on Chemical Reactors CHEMREACTOR-20.
URL <https://www.sciencedirect.com/science/article/pii/S1385894713011418>
- Enikeeva, L., Faskhutdinov, A., Koledina, K., Faskhutdinova, R., Gubaydullin, I., 07 2021. Modeling and optimization of the catalytic isomerization of the pentane-hexane fraction with maximization of individual high-octane components yield. *Reaction Kinetics, Mechanisms and Catalysis* 133.
- Fathy, Nasser, D. A., Soliman, A., M., 2019. A multi-response optimization for isomerization

of light naphtha. *Chemical Engineering Journal* 79.

URL https://buescholar.bue.edu.eg/chem_eng/79

Fritzson, P., 2013. Principles of Object-Oriented Modeling and Simulation with Modelica 2.1. Wiley-IEEE press.

Nelder, J. A., Mead, R., 01 1965. A Simplex Method for Function Minimization. *The Computer Journal* 7 (4), 308–313.

URL <https://doi.org/10.1093/comjnl/7.4.308>

Powell, M. J. D., 1994. A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation. Springer Netherlands, pp. 51–67.

URL <https://app.dimensions.ai/details/publication/pub.1046127469>

Smith, J., Ness, H. C. V., Abbott, M., 2005. Introduction to Chemical Engineering Thermodynamics. McGraw-Hill Education.

Valavarasu, G., Sairam, B., 2013. Light naphtha isomerization process: A review. *Petroleum Science and Technology* 31 (6), 580–595.

URL <https://doi.org/10.1080/10916466.2010.504931>

Appendix A

Source Codes

A.1 OpenModelica Code

```
package ISOM
model hysys_comparision
  constant Chemsep_Database.Npentane comp1;
  constant Chemsep_Database.Isopentane comp2;
  constant Chemsep_Database.Nhexane comp3;
  constant Chemsep_Database.Twomethylpentane comp4;
  constant Chemsep_Database.Threemethylpentane comp5;
  constant Chemsep_Database.TwoTwodimethylbutane comp6;
  constant Chemsep_Database.TwoThreedimethylbutane comp7;
  constant Chemsep_Database.Cyclohexane comp8;
  constant Chemsep_Database.Benzene comp9;
  constant Chemsep_Database.Hydrogen comp10;
  constant Chemsep_Database.Methylcyclopentane comp11;
  constant Chemsep_Database.Cyclopentane comp12;
  constant Chemsep_Database.Nbutane comp13;
  constant Chemsep_Database.Isobutane comp14;
  constant Chemsep_Database.Propane comp15;
  constant Chemsep_Database.Ethane comp16;
  constant Chemsep_Database.Methane comp17;

  constant Integer n = 17 "no. of components";
  constant Integer rxns = 54 "no. of reactions";
  constant Chemsep_Database.General_Properties comp[n] = {comp1, comp2,
```

```

comp3, comp4, comp5, comp6, comp7, comp8, comp9, comp10, comp11, comp12,
comp13, comp14, comp15, comp16, comp17} "comp contains all the components data ";
constant Real Kij[n, n] = {{0, 0.06, 0, 0, 0, 0, 0.0037, 0.0189, 0, 0, 0, 0,
0.0174, 0, 0.0267, 0.0078, 0.023},
{0.06, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.011, 0, -0.0056},
{0, 0, 0, 0, 0, 0, -0.003, 0.0089, -0.03, 0, 0, -0.056, 0, 0.0007, -0.04,
0.04},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0.037, 0, -0.003, 0, 0, 0, 0, 0.0126, 0, 0, 0, 0, 0, 0.0178, 0.0389}, {0.0189, 0, 0.0089, 0, 0, 0, 0, 0.0126, 0, 0, 0, 0, 0, 0.0233, 0.0322, 0.087}, {0, 0, -0.03, 0, 0, 0, 0, 0, 0, 0, 0, -0.397, 0, -0.1311, -0.0756, 0.0263}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0.0174, -0.056, 0, 0, 0, 0, 0, 0, -0.397, 0, 0, 0, -0.0004, 0.0033, 0.0089, 0.0244}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -0.0004, 0, -0.0078, -0.0067, 0.0256}, {0.02677, 0.0111, 0.0007, 0, 0, 0, 0, 0.0233, -0.1311, 0, 0, 0.0033, -0.0078, 0, 0.0011, 0.0119}, {0.0078, 0, -0.04, 0, 0, 0, 0.0178, 0.0322, -0.0756, 0, 0, 0.0089, -0.0067, 0.0011, 0, -0.003}, {0.023, -0.0056, 0.04, 0, 0, 0, 0.0389, 0.0807, 0.0263, 0, 0, 0.0244, 0.0256, 0.0119, -0.0033, 0}} "Binary Interaction coefficients";
constant Real Hf0[n](each unit = "J/mol") = {-146711.6263, -153649.33, -167200, -174300, -171600, -185600, -177800, -124600, 82900, 0, -106000, -77341.6263, -125600, -134200, -104780, -83820, -74520} "Heat of formation of nC5 and iC5 respectively at standard conditions";
constant Real Z_0[n] = {0.1491, 0.15, 0.168, 0.17, 0.167, 0.1617, 0.161, 0.1334, 0.1123, 1.01, 0.1394, 0.1146, 0.1231, 0.1269, 0.9833, 0.991, 1.0} "compressibility factor at standard state";
constant Real w[n] = comp.AF "ascentric factor";

parameter Real K1[rxns](each unit = "1/hr") = /*1*/4.76452E+18, /*2*/7.21139E+18, /*3*/33730214071.0, /*4*/41497229004.0, /*5*/1.04236E+21, /*6*/2.93778E+21, /*7*/1.34282E+19, /*8*/1.34282E+19, /*9*/2.50046E+14, /*10*/7.04724E+15, /*11*/99544918.98, /*12*/8091314788.0, /*13*/4.14972E+13, /*14*/5.72821E+14, /*15*/3.87275E+12, /*16*/3.87275E+13, /*17*/2177805.536, /*18*/7379366.791, /*19*/2.5587E+30, /*20*/8.67E+25, /*21*/2.50046E+26, /*22*/8.67E+27, /*23*/1264791.964, /*24*/547040.0177, /*25*/6.42716E+12, /*26*/5861639.3970, /*27*/1.28239E+13, /*28*/1.77019E+08, /*29*/6.28086E+17, /*30*/1.31226E+10, /*31*/5.4704E+16, /*32*/8472646.7060, /*33*/1.98618E+14, /*34*/867000.0, /*35*/6.73006E+20, /*36*/6.13789E+13, /*37*/4.44651E+15, /*38*/2.80556E+14, /*39*/2.6183E+10, /*40*/1.31226E+25, /*41*/1.28239E+30, /*42*/1.50667E+10, /*43*/1.72989E+10, /*44*/5.4704E+09, /*45*/6.13789E+24, /*46*/6.13789E+12, /*47*/6.13789E+24, /*48*/7.37937E+26, /*49*/3.87275E+30, /*50*/25586.984, /*51*/68868.25795, /*52*/21778.05536, /*53*/5.86164E+27,

```

```

/*54*/5.99817E+19}//,
parameter Real K0[rxns] = K1*5000/897;

parameter Real E[rxns](each unit = "J/mol")={/*1*/148.93, /*2*/154.28,
/*3*/143.17, /*4*/151.41, /*5*/150.98, /*6*/155.92,
/*7*/152.96, /*8*/149.95, /*9*/127.28, /*10*/139.07, /*11*/64.5,
/*12*/77.06, /*13*/146.14, /*14*/160.28, /*15*/98.28,
/*16*/105.4, /*17*/3.51, /*18*/4.79, /*19*/180.2, /*20*/400.43,
/*21*/187.05, /*22*/300.79, /*23*/51.08, /*24*/341.89,
/*25*/129.75, /*26*/88.64, /*27*/135.45, /*28*/129.29, /*29*/154.54,
/*30*/98.63, /*31*/150.29, /*32*/102.35, /*33*/168.13,
/*34*/90.7, /*35*/177.32, /*36*/222.9, /*37*/59.8, /*38*/54.08,
/*39*/330.28, /*40*/329.06, /*41*/284.97, /*42*/166.32,
/*43*/165.82, /*44*/112.05, /*45*/265, /*46*/264, /*47*/263.5,
/*48*/265, /*49*/295.62, /*50*/295, /*51*/295.19, /*52*/294,
/*53*/294.22, /*54*/278.81}//;
parameter Real Fi(unit = "mol/hr") = 5653 * 1000;
parameter Real y[n] ={3.44158838439661e-002, 1.17810649533234e-002, 2.99786839696041e-002,
3.31194760511754e-002, 2.59909367199687e-002, 1.23514295342689e-003, 6.66977194850522e-003,
1.48157352851034e-003, 2.06354682057886e-003, 0.804801890337508, 9.43126737760825e-003,
2.40659642577503e-003, 2.61613171014335e-004, 9.15646098550173e-004, 2.13795099740668e-003,
5.91655592093200e-003, 2.73923988821466e-002};
parameter Real yi[n] = y / sum(y);
parameter Real Cal[n] = Fi * yi / 6000;
parameter Real S = 6000;
// parameter Real K0[rxns] = k0*897;
constant Integer reac1[rxns] = {1, 2, 3, 4, 3, 5, 4, 5, 4, 6, 4, 7, 5, 6,
5, 7, 6, 7, 9, 8, 9,
11, 8, 11, 3, 8, 4, 11, 5, 11, 6, 11, 7, 11, 12, 1, 13, 14, 1, 1, 2, 3, 3,
3, 4, 4, 4, 4, 5, 5, 6, 6, 7};
constant Integer prod1[rxns] = {2, 1, 4, 3, 5, 3, 5, 4, 6, 4, 7, 4, 6, 5,
7, 5, 7, 6, 8, 9, 11,
9, 11, 8, 8, 3, 11, 4, 11, 5, 11, 6, 11, 7, 1, 12, 14, 13, 15, 15, 14, 15,
1, 13, 14, 2, 1, 15, 2, 13, 1, 2, 14, 2};
parameter Real P(unit = "Pa") = 3.2e+6 "inlet stream pressure";
parameter Real Ti(unit = "K") = 147 + 273.15 "inlet temperature";
parameter Real mi(each unit = "gm/hr") = Fi * yi * comp.MW "mass flowrate";
parameter Real M(unit = "gm/hr") = sum(mi) "total mass flowrate";
parameter Real xi = mi / M "initial wt fraction";
constant Real T0(unit = "K") = 298.15 "Standard Temp";
constant Real Pi = 3.141592654;
constant Real R(unit = "J/mol-K") = 8.314;
parameter Real dia = 2.5;
parameter Real ACS(unit = "m2") = 3.14*(dia^2)/4;
Real K[rxns](each unit = "mol/hr-m3reactor");
Real r[rxns](each unit = "mol/hr-m3) "rate";
Real Ca[n](each unit = "mol/m3) "concentration";
Real T(unit = "K") "temperature";
Real Q[rxns](each unit = "J/mol") "heat evolved from reaction";
Real denm_new;
Real y_i[n] "mole fraction";
Real x_i[n] "wt fraction";
Real m_i[n](each unit = "g/hr") "individual mass flowrate";
Real f_i[n](each unit = "mol/hr") "molar flow rate";

```

```

Real F(unit = "mol/hr") "overall flowrate";
Real rhom(unit = "mole/m3") "molar density";
// Real S(unit = "m3/hr") "overall Vol Flowrate";
Real Cpig[n](each unit = "J/mol-K") "heat capacity of ideal components";
Real Cp0ig[n](each unit = "J/mol-K");
Real Cp_T1[n](each unit = "J/mol-K");
Real Cp_T2[n](each unit = "J/mol-K");
Real delH_ig[n](each unit = "J/mol");
Real delH[rxns](each unit = "J/mol");
Real delH_res_1[n](each unit = "J/mol");
Real delHf0[rxns](each unit = "J/mol");
Real Tr[n], Pr[n], V[n], a[n], b[n], c[n], Z[n](each start = 1), A[n], B[n], Coeff[n, 4];
Real am, bm, Zm, Am, Bm, Coeffm[4], Vm, Cpmg, Cpresm, Cpm, dadt[n], dadt_m;
Real a_0[n], b_0[n], dadt_0[n], V_0[n];
Real r_i[rxns], Cpm_kg, T_in_c(unit = "C");
Real f[n], s;
initial equation
Ca = Ca1;
T = Ti;
equation
// reduced pressure and reduced temperature
Pr = P ./ comp.Pc;
Tr = T ./ comp.Tc;
// calc of mole-fraction
y_i = Ca / sum(Ca);
// calc of wt-fraction
x_i = y_i .* comp.MW / sum(y_i .* comp.MW);
// calc of individual mass flowrate
m_i = x_i * M;
// calc of individual molar flow rate
f_i = m_i ./ comp.MW;
// calc of overall flowrate
F = sum(f_i);
// calc of molar density of stream
rhom = P / (R * T * Zm);
// calc of overall volumetric flowrate
s = F / rhom;
// calculating rate constant
K = K0 .* exp(-E * 1000. / (R * T));
// calculating rate expression
for i in 1:rxns loop
  if i <= 18 then
    r[i] = K[i] * Ca[reac1[i]];
  end if;
end for;
r[19] = K[19] * Ca[9] * Ca[10];
r[20] = K[20] * Ca[8];
r[21] = K[21] * Ca[9] * Ca[10];
r[22] = K[22] * Ca[11];
r[23] = K[23] * Ca[8];
r[24] = K[24] * Ca[11];
r[25] = K[25] * Ca[3];
r[26] = K[26] * Ca[8] * Ca[10];
r[27] = K[27] * Ca[4];

```

```

r[28] = K[28] * Ca[11] * Ca[10];
r[29] = K[29] * Ca[5];
r[30] = K[30] * Ca[11] * Ca[10];
r[31] = K[31] * Ca[6];
r[32] = K[32] * Ca[11] * Ca[10];
r[33] = K[33] * Ca[7];
r[34] = K[34] * Ca[11] * Ca[10];
r[35] = K[35] * Ca[12] * Ca[10];
r[36] = K[36] * Ca[1];
r[37] = K[37] * Ca[13];
r[38] = K[38] * Ca[14];
r[39] = K[39] * Ca[1] * Ca[10];
r[40] = K[40] * Ca[1] * Ca[10];
r[41] = K[41] * Ca[2] * Ca[10];
r[42] = K[42] * Ca[3] * Ca[10];
r[43] = K[43] * Ca[3] * Ca[10];
r[44] = K[44] * Ca[3] * Ca[10];
r[45] = K[45] * Ca[4] * Ca[10];
r[46] = K[46] * Ca[4] * Ca[10];
r[47] = K[47] * Ca[4] * Ca[10];
r[48] = K[48] * Ca[4] * Ca[10];
r[49] = K[49] * Ca[5] * Ca[10];
r[50] = K[50] * Ca[5] * Ca[10];
r[51] = K[51] * Ca[5] * Ca[10];
r[52] = K[52] * Ca[6] * Ca[10];
r[53] = K[53] * Ca[6] * Ca[10];
r[54] = K[54] * Ca[7] * Ca[10];
// coeff calculation in the equation a*Z^3 + b*Z^2 + c*Z + d =0 and vanderwaals constant
(Coeff, Coeffm, a, b, c, am, bm, A, B, Am, Bm) = compressibility(P, comp.Pc, Pr,
T, comp.Tc, Tr, Kij, w, y_i, n);
(, , a_0, b_0) = compressibility(P, comp.Pc, Pr, T0, comp.Tc, T0 ./ comp.Tc,
Kij, w, y_i, n);
// compressibility factor calculation
for i in 1:n loop
    Coeff[i, 1] * Z[i] ^ 3 + Coeff[i, 2] * Z[i] ^ 2 + Coeff[i, 3] * Z[i] + Coeff[i, 4] = 0;
end for;
Coeffm[1] * Zm ^ 3 + Coeffm[2] * Zm ^ 2 + Coeffm[3] * Zm + Coeffm[4] = 0;
// molar volume calculation
V = R * T / P * Z;
V_0 = R * T / P * Z_0;
Vm = Zm * R * T / P;
// Modelica.Utilities.Streams.print(String(V_0));
// cp values of ideal gas at different temp between T0 and T
Cpig = Functions.VapCpId(comp.VapCp, T);
Cp0ig = Functions.VapCpId(comp.VapCp, T0);
Cp_T1 = Functions.VapCpId(comp.VapCp, T0 + (T - T0) / 3);
Cp_T2 = Functions.VapCpId(comp.VapCp, T0 + (T - T0) / 6);
// ideal cp for the entire stream
Cpigm = sum(y_i .* Cpig);
// cp residual calc
(Cpresm, dadt_m) = Cp_res_m(P, comp.Pc, Vm, T, comp.Tc, a, b, c, am, bm, Kij, y_i, n);
// cp real calc
Cpm = Cpigm + Cpresm;
// der(a) calc using Cp_res function which will be used in the delH_res calc

```

```

for i in 1:n loop
    ( , , , dadt[i] ) = Cp_res(comp[i].Pc, V[i], T, comp[i].Tc, Tr[i], w[i]);
    ( , , , dadt_0[i] ) = Cp_res(comp[i].Pc, V_0[i], T0, comp[i].Tc, T0 / comp[i].Tc, w[i]);
end for;
for i in 1:n loop
    // calculating the heat required to cool reactants ideal"
    delH_ig[i] = 1 / 3 * (T - T0) * (Cpig[i] + Cp0ig[i] + 4 * Cp_T1[i] + 2 * Cp_T2[i]);
// calc of residual H to cool reactants
    delH_res_1[i] = enthalpy_resid(a[i], b[i], Z[i], dadt[i], P, T) -
        enthalpy_resid(a_0[i], b_0[i], Z_0[i], dadt_0[i], P, T0);
end for;
for i in 1:rxns loop
    if i <= 18 then
// calc overall heat for cooling of reactants and heating of products for each reaction
        delH[i] = delH_ig[prod1[i]] - delH_ig[reac1[i]] + delH_res_1[prod1[i]] -
            delH_res_1[reac1[i]];
        end if;
// calc of heat of reaction at standard state for all the reactions
        delHf0[i] = Hf0[prod1[i]] - Hf0[reac1[i]];
    end for;
    delH[19] = delH_ig[8] + delH_res_1[8] - (3 * (delH_ig[10] + delH_res_1[10]) +
        delH_ig[9] + delH_res_1[9]);
    delH[20] = -delH[19];
    delH[21] = delH_ig[11] + delH_res_1[11] - (3 * (delH_ig[10] + delH_res_1[10]) +
        delH_ig[9] + delH_res_1[9]);
    delH[22] = -delH[21];
    delH[23] = delH_ig[11] - delH_ig[8] + delH_res_1[11] - delH_res_1[8];
    delH[24] = -delH[23];
    delH[25] = delH_ig[10] + delH_res_1[10] + delH_ig[8] + delH_res_1[8] -
        (delH_ig[3] + delH_res_1[3]);
    delH[26] = -delH[25];
    delH[27] = delH_ig[10] + delH_res_1[10] + delH_ig[11] + delH_res_1[11] -
        (delH_ig[4] + delH_res_1[4]);
    delH[28] = -delH[27];
    delH[29] = delH_ig[10] + delH_res_1[10] + delH_ig[11] + delH_res_1[11] -
        (delH_ig[5] + delH_res_1[5]);
    delH[30] = -delH[29];
    delH[31] = delH_ig[10] + delH_res_1[10] + delH_ig[11] + delH_res_1[11] -
        (delH_ig[6] + delH_res_1[6]);
    delH[32] = -delH[31];
    delH[33] = delH_ig[10] + delH_res_1[10] + delH_ig[11] + delH_res_1[11] -
        (delH_ig[7] + delH_res_1[7]);
    delH[34] = -delH[33];
    delH[35] = delH_ig[1] + delH_res_1[1] - (delH_ig[10] + delH_res_1[10] +
        delH_ig[12] + delH_res_1[12]);
    delH[36] = -delH[35];
    delH[37] = delH_ig[14] + delH_res_1[14] - (delH_ig[13] + delH_res_1[13]);
    delH[38] = -delH[37];
    delH[39] = delH_ig[15] + delH_res_1[15] + delH_ig[16] + delH_res_1[16] -
        (delH_ig[10] + delH_res_1[10] + delH_ig[1] + delH_res_1[1]);
    delH[40] = delH_ig[13] + delH_res_1[13] + delH_ig[17] + delH_res_1[17] -
        (delH_ig[10] + delH_res_1[10] + delH_ig[1] + delH_res_1[1]);
    delH[41] = delH_ig[14] + delH_res_1[14] + delH_ig[17] + delH_res_1[17] -
        (delH_ig[10] + delH_res_1[10] + delH_ig[2] + delH_res_1[2]);

```

```

delH[42] = 2 * (delH_ig[15] + delH_res_1[15]) - (delH_ig[10] +
delH_res_1[10] + delH_ig[3] + delH_res_1[3]);
delH[43] = delH_ig[17] + delH_res_1[17] + delH_ig[1] + delH_res_1[1] -
(delH_ig[10] + delH_res_1[10] + delH_ig[3] + delH_res_1[3]);
delH[44] = delH_ig[13] + delH_res_1[13] + delH_ig[16] + delH_res_1[16] -
(delH_ig[10] + delH_res_1[10] + delH_ig[3] + delH_res_1[3]);
delH[45] = delH_ig[14] + delH_res_1[14] + delH_ig[16] + delH_res_1[16] -
(delH_ig[10] + delH_res_1[10] + delH_ig[4] + delH_res_1[4]);
delH[46] = delH_ig[2] + delH_res_1[2] + delH_ig[17] + delH_res_1[17] -
(delH_ig[10] + delH_res_1[10] + delH_ig[4] + delH_res_1[4]);
delH[47] = delH_ig[1] + delH_res_1[1] + delH_ig[17] + delH_res_1[17] -
(delH_ig[10] + delH_res_1[10] + delH_ig[4] + delH_res_1[4]);
delH[48] = 2 * (delH_ig[15] + delH_res_1[15]) - (delH_ig[10] +
delH_res_1[10] + delH_ig[4] + delH_res_1[4]);
delH[49] = delH_ig[17] + delH_res_1[17] + delH_ig[2] + delH_res_1[2] -
(delH_ig[10] + delH_res_1[10] + delH_ig[5] + delH_res_1[5]);
delH[50] = delH_ig[16] + delH_res_1[16] + delH_ig[13] + delH_res_1[13] -
(delH_ig[10] + delH_res_1[10] + delH_ig[5] + delH_res_1[5]);
delH[51] = delH_ig[1] + delH_res_1[1] + delH_ig[17] + delH_res_1[17] -
(delH_ig[10] + delH_res_1[10] + delH_ig[5] + delH_res_1[5]);
delH[52] = delH_ig[17] + delH_res_1[17] + delH_ig[2] + delH_res_1[2] -
(delH_ig[10] + delH_res_1[10] + delH_ig[6] + delH_res_1[6]);
delH[53] = delH_ig[14] + delH_res_1[14] + delH_ig[16] + delH_res_1[16] -
(delH_ig[10] + delH_res_1[10] + delH_ig[6] + delH_res_1[6]);
delH[54] = delH_ig[17] + delH_res_1[17] + delH_ig[2] + delH_res_1[2] -
(delH_ig[10] + delH_res_1[10] + delH_ig[7] + delH_res_1[7]);
// heat evolved from each reaction
Q = delHf0 + delH;
// denominator in the enthalpy balance
denm_new = F / S * Cpm;
// component balance
der(Ca[1]) = ACS * (1 / S) * ((-r[1]) + r[2] + r[35] - r[36] - r[39] -
r[40] + r[43] + r[47] + r[51]);
der(Ca[2]) = ACS * (1 / S) * (r[1] - r[2] - r[41] + r[46] + r[49] +
r[52] + r[54]);
der(Ca[3]) = ACS * (1 / S) * ((-r[3]) + r[4] - r[5] + r[6] - r[25] +
r[26] - r[42] - r[43] - r[44]);
der(Ca[4]) = ACS * (1 / S) * (r[3] - r[4] - r[7] + r[8] - r[9] + r[10] -
r[11] + r[12] - r[27] + r[28] - r[45] - r[46] - r[47] - r[48]);
der(Ca[5]) = ACS * (1 / S) * (r[5] - r[6] + r[7] - r[8] - r[13] + r[14] -
r[15] + r[16] - r[29] + r[30] - r[49] - r[50] - r[51]);
der(Ca[6]) = ACS * (1 / S) * ((-r[10]) + r[9] + r[13] - r[14] - r[17] +
r[18] - r[31] + r[32] - r[52] - r[53]);
der(Ca[7]) = ACS * (1 / S) * ((-r[12]) + r[11] + r[15] - r[16] + r[17] -
r[18] - r[33] + r[34] - r[54]);
der(Ca[8]) = ACS * (1 / S) * ((-r[20]) + r[19] - r[23] + r[24] + r[25] -
r[26]);
der(Ca[9]) = ACS * (1 / S) * (r[20] - r[19] - r[21] + r[22]);
der(Ca[10]) = ACS * (1 / S) * (3*(r[20] - r[19] - r[21] + r[22]) + r[25] -
r[26] + r[27] - r[28] +
r[29] - r[30] + r[31] - r[32] + r[33] - r[34] - r[35] + r[36] - sum(r[39:54]));
der(Ca[11]) = ACS * (1 / S) * (r[21] - r[22] + r[23] - r[24] + r[27] -
r[28] + r[29] - r[30] +

```

```

r[31] - r[32] + r[33] - r[34]);
der(Ca[12]) = ACS * (1 / S) * ((-r[35]) + r[36]);
der(Ca[13]) = ACS * (1 / S) * ((-r[37]) + r[38] + r[40] + r[44] + r[50]);
der(Ca[14]) = ACS * (1 / S) * ((+r[37]) - r[38] + r[41] + r[45] + r[53]);
der(Ca[15]) = ACS * (1 / S) * (r[39] + 2 * r[42] + 2 * r[48]);
der(Ca[16]) = ACS * (1 / S) * (r[39] + r[44] + r[45] + r[50] + r[53]);
der(Ca[17]) = ACS * (1 / S) * (r[40] + r[41] + r[43] + r[46] + r[47] +
r[49] + r[51] + r[52] + r[54]);

der(T) = -ACS*(1/S)*(sum(Q.*r)/(denm_new));
when time >= 1.46 then
reinit(T,420.15);
end when;

// unit conversions
T_in_c = T - 273.15;
Cpm_kg = Cpm / sum(y_i .* comp.MW);
r_i = r * (1 / 3.6e+6);
//converting mol/m3-hr to Kmol/m3-Sec
f = 0.001 * f_i;
end hysys_comparision;

function enthalpy_resid
input Real am, bm, Zm, dadT, P, T;
output Real Hr;
protected
constant Real R = 8.314, V0 = R * 298.15 / 101325, uu = 2, ww = -1;
Real B, DAres, DSres, DHres;
algorithm
B := bm * P / (R * T);
DAres := am / (bm * (uu ^ 2 - 4 * ww) ^ 0.5) * log((2 * Zm + B *
(uu - (uu ^ 2 - 4 * ww) ^ 0.5)) /
(2 * Zm + B * (uu + (uu ^ 2 - 4 * ww) ^ 0.5)))
- R * T * log((Zm - B) / Zm) - R * T * log(Zm);
DSres := R * log((Zm - B) / Zm) + R * log(Zm) -
1 / (8 ^ 0.5 * bm) * dadT * log((2 * Zm + B * (2 - 8 ^ 0.5)) /
(2 * Zm + B * (2 + 8 ^ 0.5)));
Hr := DAres + T * DSres + R * T * (Zm - 1);
end enthalpy_resid;

function Cp_res_m
input Real P, Pc[], Vm, T, Tc[], a[], b[],
c[], am, bm, Kij[:, :], y[], n;
output Real Cpres, dadt, d2adt, dpdt, d2pdt, dvdt;
protected
constant Real R = 8.314;
Real aux1, aux2;
algorithm
aux1 := -R / 2 * (0.45724 / T) ^ 0.5;
aux2 := 0;
for i in 1:n loop
for j in 1:n loop
aux2 := aux2 + y[i] * y[j] * (1 - Kij[i, j]) *
(c[j] * (a[i] * Tc[j] / Pc[j]) ^ 0.5 + c[i] *

```

```

        (a[j] * Tc[i] / Pc[i]) ^ 0.5);
    end for;
end for;
dadt := aux1 * aux2;
d2adt := R / 4 * (0.45724 / T) ^ 0.5 * (1 / T) * aux2;
dpdt := R / (Vm - bm) - dadt / (Vm * (Vm + bm) + bm *
(Vm - bm));
d2pdt := -d2adt / (Vm * (Vm + bm) + bm * (Vm - bm));
dvdt := dpdt / (R * T / (Vm - bm) ^ 2 - am * (2 * Vm + 2 * bm)
/ (Vm * (Vm + bm) + bm * (Vm - bm)) ^ 2);
Cpres := (-R) + T * dpdt * dvdt - T * d2adt / (8 ^ 0.5 * bm)
* log((Vm + (1 - 2 ^ 0.5) * bm) / (Vm + (1 + 2 ^ 0.5) * bm));
end Cp_res_m;

function Cp_res
    input Real Pc, V, T, Tc, Tr, w;
    output Real dpdt, d2p_dt2, dpdv, dvdt, dera, der2a, Cpres;
protected
    constant Real coeff1 = 0.37464 + 1.54226 * w - 0.26992 * w ^ 2,
coeff2 = 0.45724 * ((R * Tc) ^ 2 / Pc), coeff = coeff1 * coeff2, R = 8.314;
    Real a, b;
algorithm
    a := (1 + coeff1 * (1 - Tr ^ 0.5)) ^ 2 * coeff2;
    b := 0.0778 * (R * Tc) / Pc;
    dpdv := (-R * T / (V - b) ^ 2) - a * (2 * V + 2 * b)
/ (V * (V + b) + b * (V - b)) ^ 2;
    dera := (-coeff1 * coeff2 / (Tc * T) ^ 0.5) *
(1 + coeff1 * (1 - Tr ^ 0.5));
    der2a := 0.5 * (coeff1 * coeff2 * (1 + coeff1 * (1 - Tr ^ 0.5))
/ (Tc ^ 0.5 * T ^ 1.5) + coeff1 ^ 2 * coeff2 / (Tc * T));
    dpdt := R / (V - b) - dera / (V * (V + b) + b * (V - b));
    d2p_dt2 := -der2a / (V * (V + b) + b * (V - b));
    dvdt := dpdt / (R * T / (V - b) ^ 2 - a * (2 * V + 2 * b)
/ (V * (V + b) + b * (V - b)) ^ 2);
    Cpres := (-R) + T * dpdt * dvdt - T * der2a
/ (8 ^ 0.5 * b) * log((V + (1 - 2 ^ 0.5) * b) / (V + (1 + 2 ^ 0.5) * b));
end Cp_res;

function compressiblity
    input Real P, Pc[:,], Pr[:,], T, Tc[:,], Tr[:,], Kij[:, :, :],
w[:,], y_i[:,];
    input Integer n;
    output Real Coeff[n, 4], Coeffm[4], a[n], b[n], c[n],
am, bm, A[n], B[n], Am, Bm;
protected
    constant Real R = 8.314;
algorithm
    c := 0.37464 .+ 1.54226 * w .- 0.26992 * w .^ 2;
    a := (1 .+ c .* (1 .- Tr .^ 0.5)) .^ 2 .* 0.45724 .* (R * Tc) .^ 2 ./ Pc;
    b := 0.0778 * (R * Tc) ./ Pc;
    A := a * (P / (R * T) ^ 2);
    B := b * (P / (R * T));
    Coeff[:, 1] := ones(n);

```

```

Coeff[:, 2] := B .- 1;
Coeff[:, 3] := A .- 3 * B .^ 2 .- 2 * B;
Coeff[:, 4] := B .^ 3 + B .^ 2 .- A .* B;
am := 0.0;
bm := 0.0;
for i in 1:n loop
    for j in 1:n loop
        am := am + y_i[i] * y_i[j] * sqrt(a[i] * a[j]) *
            (1 - Kij[i, j]);
    end for;
    bm := bm + y_i[i] * b[i];
end for;
Am := am * (P / (R * T) ^ 2);
Bm := bm * (P / (R * T));
Coeffm[1] := 1;
Coeffm[2] := Bm - 1;
Coeffm[3] := Am - 3 * Bm ^ 2 - 2 * Bm;
Coeffm[4] := Bm ^ 3 + Bm ^ 2 - Am * Bm;
end compressiblty;
end ISOM;

```

A.2 R-Language Code

```

# Loading libraries
library(nloptr) # For COBYLA, Nelder-Mead and Subplex
library(data.table)

# Reading the original data
original_data = read.csv("original_data.csv")
original_data = setDT(original_data)

# no. of parameters
n <- 38

objective <- function(k)
{
    # Running simulation
    system(paste0("isom.hysys_comparision.exe_override=E[1]=", k[1],
                  ",E[2]=", k[2], ",E[3]=", k[3], ",E[4]=", k[4], ",E[5]=", k[5],
                  ",E[6]=", k[6], ",E[7]=", k[7], ",E[8]=", k[8], ",E[9]=", k[9],
                  ",E[10]=", k[10], ",E[11]=", k[11], ",E[12]=", k[12], ",E[13]=", k[13],
                  ",E[14]=", k[14], ",E[15]=", k[15], ",E[16]=", k[16], ",E[17]=", k[17],
                  ",E[18]=", k[18], ",E[19]=", k[19], ",E[20]=", k[20], ",E[21]=", k[21],
                  ",E[22]=", k[22], ",E[23]=", k[23], ",E[24]=", k[24], ",E[25]=", k[25],
                  ",E[26]=", k[26], ",E[27]=", k[27], ",E[28]=", k[28], ",E[29]=", k[29],
                  ",E[30]=", k[30], ",E[31]=", k[31], ",E[32]=", k[32], ",E[33]=", k[33],
                  ",E[34]=", k[34], ",E[35]=", k[35], ",E[36]=", k[36], ",E[37]=", k[37],
                  ",E[38]=", k[38], ",E[39]=", k[39], ",E[40]=", k[40], ",E[41]=", k[41],
                  ",E[42]=", k[42], ",E[43]=", k[43], ",E[44]=", k[44], ",E[45]=", k[45],

```

```

" ,E[46]="" ,k[46], " ,E[47]="" ,k[47], " ,E[48]="" ,k[48], " ,E[49]="" ,k[49],
" ,E[50]="" ,k[50], " ,E[51]="" ,k[51], " ,E[52]="" ,k[52], " ,E[53]="" ,k[53],
" ,E[54]="" ,k[54]))
```

Reading data

```

simulated_data <- fread("ISOM.hysys_comparision_res.csv",
                         select = c("x_i[1]", "x_i[2]", "x_i[3]", "x_i[4]", "x_i[5]",
                         "x_i[6]", "x_i[7]", "x_i[8]", "x_i[11]"))[, c(2,3,6)]
error = sum(abs(simulated_data - original_data))
cat("Error_value_is:", error, "\n")
return(error)
}
```

initial_par = c(148.93, 154.28, 143.17, 151.41,
150.98, 155.92, 152.96, 149.95,
127.28, 139.07, 64.5, 77.06,
146.14, 160.28, 98.28, 105.4,
3.51, 4.79, 180.2, 400.43,
187.05, 300.79, 51.08, 341.89,
129.75, 88.64, 135.45, 129.29,
154.54, 98.63, 150.29, 102.35,
168.13, 90.7, 177.32, 222.9,
59.8, 54.08, 330.28, 329.06,
284.97, 166.32, 165.82, 112.05,
265, 264, 263.5, 265,
295.62, 295, 295.19, 294,
294.22, 278.81)

lower_bound = initial_par - 0.7 * initial_par
upper_bound = initial_par + 0.7 * initial_par

opt=cobyla(x0 = initial_par,
 fn = objective,
 lower = lower_bound,
 upper = upper_bound,
 nl.info=TRUE,
 control = list(xtol_rel= 1e-6, maxeval=3000))

A.3 Reactions

No	Reaction	E, kJ/mol	lg(k ₀)	No	Reaction	E, kJ/mol	lg(k ₀)
1	n-C ₅ H ₁₂ →iso-C ₅ H ₁₂	148.93	15.74	28	MCP+H ₂ →2-MP	129.29	8.31
2	iso-C ₅ H ₁₂ →n-C ₅ H ₁₂	154.28	15.92	29	3-MP→MCP+H ₂	154.54	14.86
3	n-C ₆ H ₁₄ →2-MP	143.17	7.59	30	MCP+H ₂ →3-MP	98.63	10.18
4	2-MP→n-C ₆ H ₁₄	151.41	7.68	31	2,2-DMB→MCP+H ₂	150.29	13.80
5	n-C ₆ H ₁₄ →3-MP	150.98	18.08	32	MCP+H ₂ →2,2-DMB	102.35	6.99
6	3-MP→n-C ₆ H ₁₄	155.92	18.53	33	2,3-DMB→MCP+H ₂	168.13	11.36
7	2-MP→3-MP	152.96	16.19	34	MCP+H ₂ →2,3-DMB	90.70	6.00
8	3-MP→2-MP	149.95	16.19	35	CP+H ₂ →n-C ₅ H ₁₂	177.32	20.89
9	2-MP→2,2-DMB	127.28	11.46	36	n-C ₅ H ₁₂ →CP+H ₂	222.90	10.85
10	2,2-DMB→2-MP	139.07	12.91	37	n-C ₄ H ₁₀ →iso-C ₄ H ₁₀	59.80	12.71
11	2-MP→2,3-DMB	64.50	5.06	38	iso-C ₄ H ₁₀ →n-C ₄ H ₁₀	54.08	11.51
12	2,3-DMB→2-MP	77.06	6.97	39	n-C ₅ H ₁₂ +H ₂ →C ₃ H ₈ +C ₂ H ₆	330.28	10.48
13	3-MP→2,2-DMB	146.14	10.68	40	n-C ₅ H ₁₂ +H ₂ →n-C ₄ H ₁₀ +CH ₄	329.06	25.18
14	2,2-DMB→3-MP	160.28	11.82	41	iso-C ₅ H ₁₂ +H ₂ →iso-C ₄ H ₁₀ +CH ₄	284.97	30.17
15	3-MP→2,3-DMB	98.28	9.65	42	n-C ₆ H ₁₄ +H ₂ →2 C ₃ H ₈	166.32	10.24
16	2,3-DMB→3-MP	105.40	10.65	43	n-C ₆ H ₁₄ +H ₂ →n-C ₅ H ₁₂ +CH ₄	165.82	10.30
17	2,2-DMB→2,3-DMB	3.51	3.40	44	n-C ₆ H ₁₄ +H ₂ →n-C ₄ H ₁₀ +C ₂ H ₆	112.05	9.80
18	2,3-DMB→2,2-DMB	4.79	3.93	45	2-MP+H ₂ →iso-C ₄ H ₁₀ +C ₂ H ₆	265.00	24.85
19	B+3H ₂ →CH* (cyclohexane)	180.20	30.47	46	2-MP+H ₂ →iso-C ₅ H ₁₂ +CH ₄	264.00	12.85
20	CH* (cyclohexane)→B+3H ₂	400.43	23.00	47	2-MP+H ₂ →n-C ₅ H ₁₂ +CH ₄	263.50	24.85
21	B+3H ₂ →MCP	187.05	26.46	48	2-MP+H ₂ →2 C ₃ H ₈	265.00	26.93
22	MCP→B+3H ₂	300.79	25.00	49	3-MP+H ₂ →iso-C ₅ H ₁₂ +CH ₄	295.62	30.65
23	CH* (cyclohexane)→MCP	51.08	3.164	50	3-MP+H ₂ →n-C ₄ H ₁₀ +C ₂ H ₆	295.00	4.47
24	MCP→CH* (cyclohexane)	341.89	2.80	51	3-MP+H ₂ →n-C ₅ H ₁₂ +CH ₄	295.19	4.90
25	n-C ₆ H ₁₄ →CH (cyclohexane*)+H ₂	129.75	9.87	52	2,2-DMB+H ₂ →iso-C ₅ H ₁₂ +CH ₄	294.00	4.40
26	CH* (cyclohexane)+H ₂ →n-C ₆ H ₁₄	88.64	6.83	53	2,2-DMB+H ₂ →iso-C ₄ H ₁₀ +C ₂ H ₆	274.22	27.83

Figure A.1: The reactions considered in this work (Enikeeva et al., 2021)

Appendix B

Interoperability of DWSIM and Open Modelica

B.1 Concept

I've also tried to integrate DWSIM and Open Modelica. DWSIM is an opensource chemical simulator that offers the builtin thermodynamic packages, CAPE-OPEN ability, UI for drag and drop of unit operation and also custom modeling of unit operations. With the help of custom modeling users can write their own model in two languages namely 1) Iron python and 2) python.NET. Refer to unit operation 'PfrOM' in Figure B.1 it is a custom model. This is how it works:

- In the backend of this unit operation(PfrOM) a python script is written in such a way that it takes the data(temp, pressure, molar flow, composition) from the 'LN' stream.
- This data is used by the model to produce results.
- These results are then passed on to 'prod' stream by the python code only.

- Later on stream 'prod' is connected to the inbuilt unit operations conventionally.

The idea presented in the preamble of this chapter will be helpful to integrate DWSIM and Open Modelica. The figure-B.1 and the code in appendix-B.2 shows the integration of Open Modelica and DWSIM. The reason this method is chosen because while defining the material stream, DWSIM offers nice User Interface and it is easy to work with than the code.

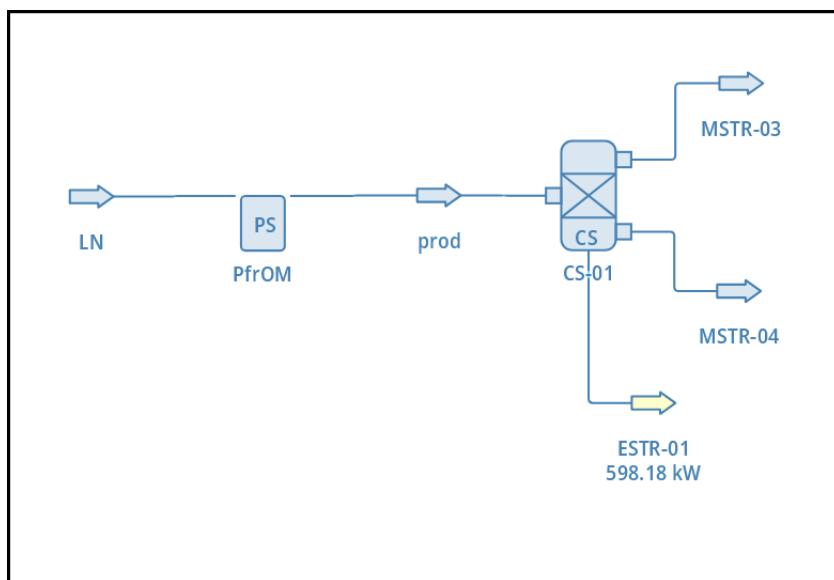


Figure B.1: Figure showing the usage of custom Modeling in DWSIM

B.2 Source Code

```

import clr
import os
import pandas as pd
clr.AddReference("DWSIM.Thermodynamics")

#ims1 is the material stream and we are loading into the feed variable
feed = ims1

#Extracting the Temp, pressure, molar_flow_rate and mole fraction from feed variable
Nc = len(feed.ComponentIds)
feed_mol_frac = feed.GetProp("fraction", "Overall", None, "", "mole")
T = feed.GetProp("temperature", "Overall", None, "", "")[0]
P = feed.GetProp("pressure", "Overall", None, "", "")[0]
F = feed.GetProp("totalflow", "Overall", None, "", "mole")[0]

```

```

f_vap = feed.GetProp("flow", "vapor", None, "", "mole")
vol_vap = feed.GetProp("volume", "vapor", None, "", "")

argument = "isom.hysys_comparision.exe_override="

#empty list to collect the names of components
index = []

for i in range(Nc):
    argument = argument+f"y[{i+1}]={feed_mol_frac[i]},"
    index.append(f"y_i[{i+1}]")

#system methode invokes the command line
os.system(argument)

#this method helps us in printing
Flowsheet.WriteMessage(str(argument))

#extracting the values from csv
df = pd.read_csv(r"C:/Users/pavan/AppData/Local/dwsim7/ISOM.hysys_comparision_res.csv")
mol_frac_outlet = df.loc[11,index].values.tolist()
temp_outlet = float(df.loc[11,'T'])
F_outlet = float(df.loc[11,'F_si'])

#Flowsheet.WriteMessage(str(temp_outlet))

#setting properties extracted from csv file to the omsl stream
oms1.Clear()
oms1.ClearAllProps()
oms1.SetProp("totalflow", "Overall", None, "", "mole", [F_outlet])
oms1.SetProp("temperature", "Overall", None, "", "", [temp_outlet])
oms1.SetProp("fraction", "Overall", None, "", "mole", mol_frac_outlet)
oms1.SetProp("pressure", "Overall", None, "", "", [P])

```